



**FACULTAD DE INGENIERÍA
CARRERA DE INFORMÁTICA**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO INFORMÁTICO**

**TÍTULO: “SISTEMA DE GESTIÓN DE HORARIOS DOCENTES
PARA LA UNIVERSIDAD DE CIENFUEGOS”**

Autor:

Leonardo Alain Moreira Rodríguez

Tutores:

MsC. Richard Darian Sánchez Rivero

MsC. Domingo José Valladares Pérez

Cienfuegos, Cuba

Curso: 2023

Resumen

La presente investigación está orientada al desarrollo e implementación de un sistema para la gestión de los horarios docentes en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”. Este sistema es una aplicación web que sirve de herramienta para la elaboración de los horarios docentes, permitiendo el control a los planificadores, sobre todos los elementos vinculados a su confección como los cursos académicos, los períodos, los días no disponibles, las formas de organización docente, los tipos de actividades extracurriculares, asignaturas, locales y profesores. Además, brinda a todos los estudiantes y profesores el acceso a sus horarios, organizados por facultades, carreras, años académicos y grupos; y la posibilidad de exportarlo a formato PDF y JPG. A través de este documento de investigación se describen los elementos que conforman el análisis, diseño, implementación y validación del sistema propuesto apoyándose en el uso de la metodología de desarrollo de software Proceso Unificado Ágil (AUP). Este sitio está desarrollado con el framework Django, utilizando Python como lenguaje de programación del lado del servidor con el patrón arquitectónico Modelo Template Vista (MTV), el framework Bootstrap con HTML, CSS y JavaScript del lado del cliente y PostgreSQL como sistema gestor de bases de datos.

Palabras claves: horario docente, planificadores, estudiantes, profesores, aplicación web.

Abstract

The present research is aimed at the development and implementation of a system for the management of teaching schedules at the “Carlos Rafael Rodríguez” University of Cienfuegos. This system is a web application that serves as a tool for preparing teaching schedules, allowing planners control over all the elements linked to their preparation such as academic courses, periods, unavailable days, forms of teaching organization, types of extracurricular activities, subjects, rooms and teachers. In addition, it gives all students and teachers access to their schedules, organized by faculties, majors, academic years and groups; and the possibility of exporting it to PDF and JPG format. Through this research document, the elements that make up the analysis, design, implementation and validation of the proposed system are described, based on the use of the Agile Unified Process (AUP) software development methodology. This site is developed with the Django framework, using Python as a server-side programming language with the Model Template View (MTV) architectural pattern, the Bootstrap framework with HTML, CSS and JavaScript on the client side and PostgreSQL as a database management system.

Keywords: teaching schedule, planners, students, teachers, web application.

Índice

Introducción	1
Capítulo I: “Fundamentación Teórica”	5
1.1 Introducción	5
1.2 Descripción del dominio del problema	5
1.3 Flujo actual de los procesos	6
1.4 Descripción de los sistemas existentes	7
1.5 Tendencias, metodologías y/o tecnologías actuales	10
1.6 Conclusiones.....	13
Capítulo II “Descripción y Construcción de la Solución Propuesta”	14
2.1 Introducción	14
2.2 Concepción general del sistema.....	14
2.3 Requisitos de software	14
2.4 Descripción de la arquitectura	20
2.5 Diagrama de clases.....	21
2.6 Modelo de la base de datos	22
2.7 Patrones de diseño utilizados	23
2.8 Estándar de codificación empleado.....	23
2.9 Conclusiones.....	24
Capítulo III “Validación de la Solución Propuesta”	25
3.1 Introducción	25
3.2 Métrica aplicada a los requerimientos	25
3.3 Pruebas de caja blanca.....	25
3.4 Pruebas de caja negra	29
3.5 Pruebas de aceptación	40
3.6 Conclusiones.....	41
Conclusiones.....	42
Recomendaciones.....	43
Referencias bibliográficas	44
Anexos	46
Anexo 1 Modelo oficial del P4.....	46
Anexo 2 Pruebas unitarias	46

Índice de Tablas

Tabla 1 Descripción de los requisitos funcionales del sistema.....	19
Tabla 2 Descripción de los requisitos no funcionales del sistema.	19
Tabla 3 Historia de Usuario del requisito Funcional Mostrar Horario.....	20
Tabla 4 Caso de prueba del camino básico 1	28
Tabla 5 Caso de prueba del camino básico 2	28
Tabla 6 Caso de prueba del camino básico 3	28
Tabla 7 Caso de prueba del camino básico 4	29
Tabla 8 Clases de equivalencia para el Campo Nombre.....	29
Tabla 9 Clases de equivalencia para los Campos Inicio y Fin.....	30
Tabla 10 Clases de equivalencia para los Campos Inicio del turno de clases, Entretiempo, Duración del turno de clases y Duración del receso	30
Tabla 11 Caso de prueba A de particiones de equivalencia	31
Tabla 12 Caso de prueba B de particiones de equivalencia.....	32
Tabla 13 Caso de prueba C de particiones de equivalencia	33
Tabla 14 Caso de prueba D de particiones de equivalencia	34
Tabla 15 Caso de prueba E de particiones de equivalencia.....	35
Tabla 16 Caso de prueba F de particiones de equivalencia.....	36
Tabla 17 Caso de prueba G de particiones de equivalencia	37
Tabla 18 Caso de prueba H de particiones de equivalencia	38
Tabla 19 Caso de prueba I de particiones de equivalencia	39
Tabla 20 Caso de prueba J de particiones de equivalencia	40

Índice de Figuras

Figura 1 Arquitectura Modelo Template Vista.....	21
Figura 2 Aplicación de la arquitectura a la solución propuesta.....	21
Figura 3 Diagrama de clases del diseño.....	22
Figura 4 Modelo de datos del sistema.	23
Figura 5 Vista create_faculty para agregar una Facultad.....	26
Figura 6 Grafo de flujo del código de la vista create_faculty	27

Introducción

La informática es una ciencia con aplicación en todas las esferas de la sociedad. La necesidad de softwares crece año tras año a un ritmo vertiginoso, en consecuencia, para satisfacer esta demanda, se han producido grandes avances en la industria de software.

Las aplicaciones informáticas permiten eliminar las redundancias, las inconsistencias, y garantizar la integridad y seguridad de los datos, esto tributa a tomar decisiones basadas en informaciones confiables; razón por la cual la tecnología de la información actual se impone cada día más.

Dentro de los factores claves que pueden contribuir de forma significativa a mejorar la calidad de los servicios formativos que ofrecen las universidades, encontramos la asimilación y uso apropiado de las TIC. Es por ello que toda institución educativa debe poseer una cantidad suficiente de recursos en infraestructura y medios que le permita impartir una educación de alta calidad. La existencia de dichos recursos requiere de una correcta planificación y análisis, puesto que tanto su escasez como su sobreoferta influyen negativamente en calidad de la educación. El deterioro de la educación también se pone de manifiesto cuando la planificación del proceso docente no es del agrado de los profesores y alumnos implicados en ella[1].

En Cuba, desde el año 2000 surge el Ministerio de la Informática y las Comunicaciones (MIC), impulsando el proceso de informatización de la sociedad. Se han brindado múltiples soluciones a problemas, pero desde que se ha dado las instrucciones del presidente Díaz-Canel, de aportar nosotros mismos a este proceso, ha ocurrido un auge en el surgimiento de aplicaciones, principalmente aplicaciones web y móviles, para gestionar becas, documentos, reuniones, y múltiples eventos.

Se sabe que los avances tecnológicos permiten a los estudiantes y profesores universitarios tener un amplio potencial para acceder a los servicios académicos a través de la red. La disponibilidad de nuevas herramientas accesibles desde la web, hace posible que la gestión de la información se efectúe con inmediatez, permitiendo publicar las noticias en el preciso instante en que se genera; conjuntamente, la presencia de los dispositivos móviles de última generación y una conectividad permanente a Internet, permite a todos los actores de la comunidad educativa, recoger las novedades en cualquier lugar y en cualquier instante.

En las universidades, donde se forma el potencial científico-técnico del país, a partir de un grupo de procesos denominados sustantivos, dentro de los cuales se encuentra el Proceso de Formación, guiado por el Proceso Docente Educativo (PDE), que tiene el propósito de educar, instruir y desarrollar a los futuros profesionales, sistematizando y recreando de forma planificada y organizada la cultura acumulada por la humanidad. Como parte inseparable del PDE se encuentra el Proceso de Enseñanza-Aprendizaje (PEA), que tiene carácter sistémico, organizado y es planificado por un personal capacitado con la participación de docentes y directivos de las universidades[2].

El PEA, en su organización como sistema, requiere de una distribución adecuada de las diferentes asignaturas: por semestres, semanas, días y horas lectivas. En esta organización se deben tener en cuenta las exigencias propias de las disciplinas que se imparten a los estudiantes y su importancia para la especialidad; así como las características de cada grupo de educandos, el tipo de enseñanza, la forma de organización de sus actividades docentes y las actividades extra-docentes que se realizan. Estos factores y otros afectan directamente la manera en que han de ser distribuidos los contenidos a lo largo de la formación del profesional, elementos claves para lograr una planificación balanceada de acuerdo con los principios y normas de la higiene de la actividad docente[3].

La planificación es un plan general, metódicamente organizado y frecuentemente de gran amplitud, para obtener un objetivo determinado[4]. Es una forma de organizar las tareas diarias de cualquier organización, e incluso personales. Este método responde a una necesidad del hombre: aprovechar

Sistema de Gestión de Horarios Docentes para la Universidad de Cienfuegos
eficientemente el tiempo, con el objetivo de alcanzar las metas propuestas con el menor tiempo y cantidad de recursos posibles[5].

La planificación implica un proceso consciente de estudio y selección del mejor curso de acción a seguir, estableciendo prioridades frente a una variedad de alternativas posibles y factibles de acuerdo a los recursos con que se cuenta. La planificación docente no es más que la organización coherente y funcional de un conjunto de actividades destinadas al eficaz aprendizaje de los estudiantes tratando siempre de lograr el óptimo uso de los recursos disponibles, como pueden ser los inmuebles o los recursos humanos como profesores, en un marco de tiempo adecuado.

Un problema recurrente en los centros de la enseñanza superior cubana es el tema de la confección de los horarios docentes, labor que en la mayoría de estos se realiza de forma manual y por tanto poco eficiente. Los problemas de horarios consisten esencialmente en fijar una secuencia de encuentros entre profesores y estudiantes en un período de tiempo predefinido, satisfaciendo un conjunto de restricciones de varios tipos. Dependiendo del número de recursos involucrados, la resolución de problemas de este tipo de forma manual puede tomar desde algunas horas hasta varios días y no se garantiza que los resultados obtenidos sean los más adecuados para todas las partes involucradas[6].

Automatizar los procesos de confección de horarios docentes ha sido una tarea a la que se han dedicado numerosas investigaciones. Algunos autores plantean que estos problemas no pueden ser totalmente automatizados debido a que, aunque exista información que permita afirmar que unas soluciones son mejores que otras, esta información no se expresa trivialmente a un sistema automático; además de esto, los espacios de búsqueda son normalmente muy extensos por lo que la intervención humana es fundamental para simplificarlos. No obstante, numerosas soluciones computacionales a este tipo de problemas han surgido en los últimos años. Algunas de estas aplicaciones permiten encontrar soluciones de manera automática brindando la posibilidad al usuario de retocar de forma manual los resultados obtenidos. Por otro lado, las denominadas aplicaciones semiautomáticas, ayudan a la confección de horarios docentes requiriendo una mayor intervención de los usuarios, aun así simplifican notablemente el tiempo invertido para esta tarea[6].

En las universidades cubanas se debe prestar especial cuidado a la planificación del PEA, pues en ella intervienen varios factores[7]:

- El elevado nivel de responsabilidad de los encargados de llevar a cabo la planificación, en caso de ocurrir fallos e irregularidades.
- La requerida experiencia de los especialistas para la planificación de las actividades docentes.
- La participación de profesores y directivos en la planificación, y la consiguiente necesidad de una buena comunicación entre éstos y los especialistas en dicha actividad.
- La imprescindible necesidad de mantener informados a profesores y estudiantes, como protagonistas finales de las actividades docentes, sobre la manera en que están planificadas las mismas en cada momento.
- La dinámica y flexibilidad que debe tener la planificación de las actividades docentes en dependencia de las demandas no previstas que les hace la sociedad a las universidades.

Los horarios establecidos son esenciales para optimizar el tiempo y aprovechar las oportunidades, y esto es especialmente cierto en el ámbito universitario. Los horarios docentes son la base sobre la cual se organizan las actividades académicas en las universidades, y su correcta planificación y gestión es fundamental para asegurar el cumplimiento de los objetivos educativos establecidos.

En la Universidad de Cienfuegos, la gestión de los horarios docentes se dificulta, tanto para los planificadores del horario como para los estudiantes y profesores, debido a que:

Sistema de Gestión de Horarios Docentes para la Universidad de Cienfuegos

- Cada horario de cada grupo es elaborado a papel y lápiz, los cuales son enviados en una foto a cada grupo, por lo cual en muchas ocasiones no es muy legible y en caso de cambios se repite el proceso, por lo que se hace redundante e inconsistente.
- Existen retrasos en la confección del horario docente, tanto el preliminar al inicio de cada período, como las modificaciones posteriores que ocurren por diversas causas.
- La información a los estudiantes y profesores afectados por la restructuración del horario es ineficiente, por lo que se producen cambios que no llegan a ser conocidos oportunamente.
- No existe un sitio digital donde se pueda consultar el horario docente en cualquier momento, para contribuir a minimizar las afectaciones provocadas por las deficiencias anteriores.

Siendo esto la **situación problemática** de este trabajo de diploma, por lo anterior, el **problema a resolver** es el siguiente: ¿Cómo gestionar los horarios docentes en la Universidad de Cienfuegos, de forma que facilite el trabajo a los planificadores y el acceso para los estudiantes y profesores?

El **objeto de estudio** de la investigación es el proceso de desarrollo de sistemas de gestión universitario y el **campo de acción** es la gestión de horarios docentes en la Universidad de Cienfuegos en sistemas de gestión universitaria.

Con esta investigación se quiere **defender la idea**, que con la implementación de un sistema informático para la gestión de horarios docentes se mejorará el proceso docente educativo de la Universidad, dándole mayor organización y legibilidad a este, y un acceso amplio al horario por parte de los estudiantes y profesores, obteniendo como **aporte práctico**:

Un sistema informático para la gestión de horarios en la Universidad.

Para darle solución al problema, se plantea como **objetivo general**: Desarrollar un sistema informático que contribuya a la estructuración, consistencia, persistencia, organización y legibilidad de los horarios docentes en la Universidad de Cienfuegos, desglosado en los siguientes **objetivos específicos**:

- Modelar el proceso de planificación de horarios docentes.
- Diseñar el sistema que soporte el flujo de procesos.
- Implementar el sistema informático solución al problema a resolver.
- Validar el sistema mediante pruebas funcionales.

Para cumplir los objetivos trazados se realizarán las siguientes tareas:

- Identificación de los procesos.
- Análisis del flujo actual de procesos.
- Búsqueda de sistemas informáticos actuales vinculados al campo de acción.
- Modelado del sistema.
- Generación de la base de datos.
- Construcción del sistema.
- Diseño de pruebas funcionales del sistema.
- Ejecución de pruebas funcionales del sistema.

El trabajo fue estructurado de forma que su comprensión fuese la mejor posible, separando la teoría de la práctica:

- Capítulo I “Fundamentación Teórica”:

Se hace un análisis sobre los procesos dentro del campo de acción, haciendo énfasis en los procesos que son objeto de gestión. Se realiza un estudio crítico de los sistemas existentes relacionados con el campo de acción y el campo de estudio. Se describen las tecnologías y tendencias actuales en las que se basa este trabajo.

- Capítulo II “Descripción y Construcción de la Solución Propuesta”:

Se extraen los requisitos funcionales y no funcionales. Se diseñan las historias de usuario. Se diseña la solución propuesta utilizando artefactos tales como diagramas de clases y de implementación. Se hace el diseño de la base de datos para la solución propuesta. Se definen los patrones de diseño.

- Capítulo III “Validación de la Solución Propuesta”:

Se aplican métricas para medir la calidad de la especificación de los requisitos. Se realiza el diseño de las pruebas funcionales del sistema a través de pruebas de caja blanca y caja negra. Se realizan pruebas de aceptación con el cliente.

Capítulo I: “Fundamentación Teórica”

1.1 Introducción

En este capítulo se hace una descripción general del dominio del problema y del objeto de estudio con el apoyo de las componentes de las diferentes áreas del conocimiento específico y del área de sistemas. Se realiza un análisis del objeto de estudio y la situación problémica en la que se encuentra inmerso el mismo. Se describen los sistemas existentes. Se hace una descripción de las tecnologías y tendencias actuales en las que se basa este trabajo.

1.2 Descripción del dominio del problema

En el ámbito de los sistemas de información, el dominio del problema es el conjunto de conceptos interrelacionados que es necesario conocer para entender el negocio del cliente, y, por lo tanto, para poder entender sus necesidades y proponer una solución adecuada[8].

1.2.1 Factores que inciden y participan en la planificación docente

María Onelia Chiang Molina en su libro "Higiene de la actividad docente" aborda el tema de la planificación concibiéndola como la primera función administrativa que sirve de base para muchas otras funciones, la cual determina por anticipado cuáles son los objetivos que deben cumplirse y qué debe hacerse para alcanzarlos; por tanto, es un modelo teórico para actuar en el futuro. La planificación comienza por establecer los objetivos y detallar los planes necesarios para alcanzarlos de la mejor manera posible. Determina a dónde se pretende llegar, qué debe hacerse además del cómo, cuándo y en qué orden deben suceder los acontecimientos.[3]

Se concibe dicha actividad como un proceso complejo, pues se deben tener en cuenta muchos aspectos de singular importancia referentes a la actividad docente, ya que es mediante ésta como se logra la asimilación de conocimientos científicos y la formación de habilidades correspondientes, objetivos y resultados esenciales de la propia actividad para los educandos, la cual es una forma de actividad cognoscitiva dirigida mediante el proceso de enseñanza en la escuela, regida por un conjunto de características que le dan elementos de complejidad, entre los que encontramos:

- Basada en contenidos previamente determinados en el plan de estudio de la carrera y en programas establecidos.
- Puede hacerse por bloques lectivos, ciclos o niveles, en dependencia de la carrera, el curso o el nivel de enseñanza.

1.2.2 Planificación docente en la educación superior cubana

Históricamente, en la solución a muchas situaciones y problemas organizativos a los que se ha enfrentado el hombre, ha sido de especial importancia su capacidad para planificar las actividades. El hecho de conducir un proceso se evidencia en cualquier área o sector de la sociedad, desde la producción industrial hasta la educación en todos sus niveles.

En el caso particular de la planificación docente de la Enseñanza Superior en Cuba, se debe considerar los cambios que han venido ocurriendo en ella a raíz de la Universalización, que conlleva al incremento del número de estudiantes y a la demanda de docentes, unido a la heterogénea organización de las actividades docentes, que incluye distintas modalidades tales como presenciales, semipresenciales y a distancia. Además de que no todos los años los contenidos de las asignaturas se distribuyen de la misma forma, pueden cambiar incluso el nombre de estas o unirse unas con otras, pueden cambiar el semestre o el año en que se imparten, por lo que todo esto da lugar a los diferentes planes en que puede estar dividido la carrera. Todo esto conduce a un proceso de planificación complejo, en extremo dinámico y altamente propenso a irregularidades y descoordinaciones que tienden a afectar la calidad del Proceso Docente Educativo.

1.2.3 Conceptos o elementos a considerar para la planificación de los horarios docentes

P1: Es el documento que contiene todo el contenido, las actividades docentes, el profesor, la cantidad de turnos de clase, la cantidad de horas por turno, lugar y observaciones de una asignatura.

P2: Es el documento que contiene la forma docente y las horas de una asignatura.

Malla: Es el documento que contiene todo el contenido que se impartirá en una carrera dividido por año académico de un curso académico, en este se muestran todas las asignaturas con sus horas de clases, las semanas lectivas y los períodos.

Balance de cargas: Es el documento que define los requisitos docentes, según los planes de estudio de las carreras, el fondo de tiempo declarado para el período y el número de asignaturas y semanas que lo componen para cada facultad.

Gráfico docente: Es el documento en formato Excel que muestra los meses, la fecha de inicio y fin del curso académico, con los períodos en que está dividido con fecha de inicio y fin de cada uno, con cada una de las semanas lectivas enumeradas con fecha de inicio y fin, y las semanas de receso escolar.

P4: Es el documento que contiene el horario docente de un período de un grupo de un año de una carrera de una facultad mostrándose cada uno de estos elementos, además del curso académico, el tipo de curso y una leyenda con las asignaturas y su simbología, los profesores que las imparten y las actividades y sus simbologías. [Ver Anexo 1](#)

1.3 Flujo actual de los procesos

El proceso de planificación de los horarios docentes en la Universidad de Cienfuegos se puede desglosar en varios subprocesos, de los cuales en este trabajo de diploma solo se tomarán en cuenta el tercer y cuarto subproceso, aunque de cualquier forma esto servirá para sentar las bases de la investigación, informatización e implementación de los demás subprocesos.

1.3.1 Identificación de los subprocesos

Subproceso 1: Análisis de la resolución del Ministerio de Educación Superior (MES).

- El MES emite a todas las universidades del país una resolución al final de cada curso académico, con todas las indicaciones relacionadas con el calendario académico general del siguiente curso, con los siguientes elementos:
 - Inicio del curso.
 - Fin del curso.
 - Semanas lectivas.
 - Vacaciones de verano.
 - Organización docente (cuando comienza la docencia).
 - Reinicio de las actividades docentes.
 - Semanas de receso docente, las cuales no cuentan dentro de las semanas lectivas.
- Cada universidad en función de la fecha de inicio y de fin de curso, organiza el proceso docente, en correspondencia con sus particularidades.

Subproceso 2: Elaboración de los Gráficos Docentes.

- La dirección de formación de la universidad emite a todas las facultades las indicaciones relacionadas con el calendario académico con las particularidades de la universidad y un margen en dependencia de las especificidades de cada facultad.
- Se elaboran los gráficos docentes de cada carrera.
- Se entrega la documentación necesaria (P1/P2, Mallas, Planes de impartición, Balance de Carga) por parte de los jefes de carreras y los jefes de departamentos al departamento de planificación.

Subproceso 3: Elaboración del horario docente.

- Elaboración y validación del horario de actividades docentes.
- Publicación del horario de actividades docentes.

Subproceso 4: Reelaboración del horario docente.

- Actualización del horario de actividades docentes para reprogramar o incorporar actividades que pueden surgir durante el curso académico y que en ocasiones afectan el gráfico docente.
- Publicación y notificación a estudiantes y profesores de los cambios realizados en el horario de actividades docentes.

1.3.2 Descripción detallada de los subprocesos 3 y 4.

El proceso de planificación y publicación de los horarios docentes en la Universidad de Cienfuegos tiene sus peculiaridades, actualmente se lleva a cabo de forma manual.

Elaboración del horario docente.

El departamento de planificación docente elabora los horarios a papel y lápiz. Una vez finalizados, los horarios se publican en cada facultad y los grupos de estudiantes deben buscarlos físicamente en las pizarras informativas o recibirlos mediante fotografías enviadas por los profesores a través de WhatsApp, lo que puede generar problemas de calidad y legibilidad, y es susceptible a cambios.

Los horarios publicados están diseñados para los grupos de estudiantes, y muestran en forma de leyenda, las asignaturas que se impartirán en cada turno de clase, la ubicación de las aulas, los profesores que imparten cada asignatura, el local, las actividades (Conferencias, Seminarios, Laboratorios, Talleres, etc.), actividades extracurriculares, y además deben estar presentes los días o semanas que por determinados motivos no están disponibles en el horario pero si forman parte del período de clases.

Reelaboración del horario docente.

Luego de elaborados todos los horarios docentes, en el transcurso del periodo pueden existir cambios por determinadas situaciones ajenas a los planificadores. Ejemplo de ellas son:

- Bajas de profesores.
- Incorporación o reincorporación de profesores.
- Días o semanas no disponibles que no se habían tomado en cuenta anteriormente.

Todo esto requiere de más tiempo y esfuerzo por parte de los planificadores, que en muchos casos se repite el subproceso 3.

A la hora de notificar e informar a los estudiantes y profesores de los nuevos cambios en muchas ocasiones no se hace a tiempo y en otras nunca se llega a informar.

1.4 Descripción de los sistemas existentes

En el ámbito internacional, nacional y local ha habido varios esfuerzos tanto de instituciones como universidades, en tesis de maestría, de grado y trabajos de investigación, para tratar de dar solución al problema de los horarios docentes de universidades y de escuelas en general. Es importante aclarar que cada una de estas soluciones fueron ideadas ajustándose al lugar, tiempo o necesidades en que fueron realizadas. En este acápite se muestran varios ejemplos, que son parte de búsquedas e investigaciones realizadas.

1.4.1 Internacional

En el ámbito internacional hay varios softwares que implementan soluciones al problema en cuestión, pero estos no se ajustan a las necesidades de la Universidad de Cienfuegos. Estos son:

1.4.1.1 SchoolTool

SchoolTool es una colección de software libre administrativo para escuelas. Ya que se puede instalar fácilmente y utilizar sin derechos de licencia, SchoolTool puede ser utilizado por las escuelas para un solo propósito, por los profesores particulares o en pequeños grupos en las escuelas, o en la totalidad del sistema escolar ya que tiene una completa información de los estudiantes, que abarca la demografía, libros de calificaciones, la asistencia, calendarios e informes. También es utilizado para crear informes provisionales sobre todos los estudiantes por la característica de hoja de informe que facilita la recogida de las calificaciones, resultados de exámenes, comentarios y otros datos personalizados de los maestros, a intervalos regulares [9].

1.4.1.2 UniTime

UniTime es un sistema integral de planificación educativa que apoya el desarrollo del curso y los horarios de exámenes, maneja cambios en la gestión de estos horarios, comparte el local con otros eventos, y programa las clases individuales para estudiantes. Es un sistema distribuido que permite a múltiples universidades y jefes de departamentos coordinar los esfuerzos para construir y modificar un horario que se ajuste a sus necesidades organizativas diversas al tiempo que permite la minimización de los conflictos del curso de los estudiantes. Puede ser utilizado solo para crear y mantener un horario escolar de clases y/o exámenes, o interconectado con un sistema de información estudiantil existente [10].

El sistema fue desarrollado originalmente como un esfuerzo de colaboración por profesores, estudiantes y personal de las universidades de América del Norte y Europa. El software libre se distribuye bajo una licencia de código abierto (GNU General Public License), con la esperanza de que otras universidades puedan beneficiar a sus estudiantes a través de una mejor planificación o desea contribuir a la investigación en curso en esta área [10].

1.4.2 Nacional

En Cuba se ha abordado el tema en muchas ocasiones ya que está siempre presente en diversas instituciones; principalmente en las universidades, de aquí que sean estas las más beneficiadas con la solución a este problema pero que no se ajustan a las necesidades específicas de nuestra Universidad.

1.4.2.1 Sistema para la Gestión del P4 de la Facultad de Ingeniería Eléctrica de La Universidad de Oriente

El objetivo general del sistema fue diseñar e implementar una aplicación Web para apoyar la gestión de la planificación docente.

Deficiencias:

- La visualización del P4 no se hace de la mejor forma ya que se le asignan colores a las asignaturas haciendo que los estudiantes tengan dificultad para asociar los colores que le fue asignado a cada una de las asignaturas y que se muestran en la leyenda, existiendo la posibilidad de que una modificación en el Plan de Estudio podría cambiar la asociación de los colores respecto a las asignaturas.
- No permite exportar los horarios.
- Dificultades para imprimir los horarios, ya que si no es en una impresora a color no se podrían distinguir las asignaturas.
- No se evidencia que el sistema cuente con un algoritmo de para la confección de los horarios.

- De manera general el trabajo tiene pocas referencias bibliográficas.

1.4.2.2 Sistema de Publicación de Horarios de la Universidad de Ciencias Informáticas (UCI)

Este sistema funciona filtrando la información solicitada por el usuario, o sea se selecciona la Facultad a la cual pertenece el usuario y a continuación se selecciona el horario que se desea teniendo en cuenta los diferentes criterios de búsqueda:

- Buscar horario por profesor.
- Buscar horario por grupo.
- Buscar horario por año.
- Buscar horario por brigada.
- Buscar horario por actividad.

Deficiencias:

- No se permite exportar el horario, evitando portabilidad del P4.
- No contiene la opción de imprimir.
- La forma en que se visualizan los horarios está sobrecargada de información.

1.4.2.3 Aplicación Web para la confección de Horarios Docentes

Este sistema es un sitio Web que forma parte de un proyecto de tesis de grado de la UCI. Este solo satisface las necesidades de dicha Universidad, por lo que es incompatible con la Universidad de Cienfuegos [6].

1.4.3 Local

Mediante investigaciones y entrevistas realizadas al personal que se encarga de planificar los horarios se ha podido verificar y comprobar que la planificación de estos se hace de forma manual y en poca medida apoyándose en herramientas de trabajo tales como Microsoft Excel y el software QBalance.

1.4.3.1 QBalance

QBalance es una aplicación de escritorio para la solución del problema de balance de carga docente, desarrollado en la Universidad de Cienfuegos en un proyecto de tesis de maestría del autor Boris Pérez Cañedo [11].

1.4.3.2 Sistema para la Gestión y Visualización de Horarios Docentes

Este sistema es un proyecto de tesis de grado del autor Renier Miranda Hernández. El objetivo de este es la gestión visualización y portabilidad de los horarios docentes en la Universidad de Cienfuegos [12].

Algunas de sus funcionalidades son:

- Permite exportar en formato PDF los horarios semestrales de estudiantes y profesores.
- Permite almacenar los tiempos de clases u horas de no disponibilidad de profesores, grupos o aulas.
- Permite crear un reporte interactivo para el análisis de la carga docente.

Deficiencias:

- No hay roles para estudiantes y profesores.
- No hay control de cada curso.

Sistema de Gestión de Horarios Docentes para la Universidad de Cienfuegos
Como resultado del análisis de estos sistemas o herramientas, estas no cumplen con los requisitos y especificaciones que demanda el cliente, justificando así el objetivo de la presente investigación.

1.5 Tendencias, metodologías y/o tecnologías actuales

En esta sección se describen las metodologías, lenguajes, herramientas y tecnologías utilizados para realizar este trabajo.

En el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software.

1.5.1 Agile Unified Process (AUP)

Agile Unified Process, en español Proceso Unificado Ágil de Scott Ambler o (AUP) en inglés, es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP [13].

1.5.1.1 Características de AUP.

- Versión simplificada de la metodología RUP [13].
- Abarca siete flujos de trabajos, cuatro ingenieriles y tres de apoyo: Modelado, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de Proyectos y Ambiente [13].
- El modelado agrupa los tres primeros flujos de RUP (Modelamiento del negocio, Requerimientos y Análisis y Diseño) [13].
- Dispone de cuatro fases igual que RUP: Incepción o Creación, Elaboración, Construcción y Transición [13].

1.5.1.2 Descripción de los flujos de trabajo ingenieriles.

- El Modelado es el flujo de trabajo que tiene el objetivo de entender el negocio de la organización, el problema de dominio que se aborda en el proyecto y determinar una solución viable para resolver el problema de dominio [13].
- El flujo de trabajo Implementación tiene como objetivo transformar su modelo en código ejecutable y realizar un nivel básico de las pruebas, en particular, la unidad de pruebas [13].
- El flujo de trabajo de Prueba tiene como objetivo realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, verificando que se cumplan los requerimientos [13].
- Por último, dentro de los flujos de trabajo ingenieriles se tiene el Despliegue, cuyo objetivo es el plan para la prestación del sistema y la ejecución de dicho plan, para que el sistema quede a disposición de los usuarios finales [13].

1.5.1.3 Descripción de las fases del ciclo de desarrollo.

- Incepción: identificación del alcance y dimensión del proyecto, propuesta de la arquitectura y del presupuesto del cliente [13].
- Elaboración: Confirmación de la idoneidad de la arquitectura [13].
- Construcción: Desarrollo incremental del sistema, siguiendo las prioridades funcionales de los implicados [13].
- Transición: Validación e implantación del sistema [13].

1.5.1.4 Técnicas ágiles que aplica AUP.

- Desarrollo Dirigido por Pruebas (Test Driven Development - TDD) [13].

- Modelado Ágil [13].
- Gestión de Cambios Ágil [13].
- Refactorización de Base de Datos para mejorar la productividad [13].

1.5.1.5 Principios en los que se basa AUP.

Simplicidad: Todo se describe concisamente utilizando poca documentación, no miles de ellas [13].

Agilidad: El ajuste a los valores y principios de La Alianza Ágil [13].

Centrarse en actividades de alto valor: La atención se centra en las actividades que en realidad lo requieren, no en todo el proyecto [13].

Herramienta de la independencia: Usted puede usar cualquier conjunto de herramientas que desea con el AUP. Se sugiere utilizar las herramientas más adecuadas para el trabajo, que a menudo son las herramientas simples o incluso herramientas de código abierto [13].

Usted querrá adaptar este producto para satisfacer sus propias necesidades: La metodología AUP es un producto de fácil uso utilizando cualquier herramienta. No es necesario comprar una herramienta especial, o tomar un curso, para adaptar esta metodología [13].

1.5.2 Lenguaje de Modelado Unificado (UML).

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema [14].

Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo [14].

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones [14]:

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender [14].
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción [14].
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados [14].
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión [14].

1.5.3 Patrón de diseño Modelo Template Vista (MTV).

El Modelo Template (Plantilla) Vista es un patrón de arquitectura de software utilizado por el framework Django. Es una variante del Modelo Vista Controlador (MVC) [15].

Modelo: Es la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene y la relación entre los datos [15].

Vista: Es la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: es un puente entre el modelo y la plantilla [15].

Template: Llamada también plantilla, es la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación [15].

1.5.4 Visual Paradigm 16.2

Sistema de Gestión de Horarios Docentes para la Universidad de Cienfuegos

Visual Paradigm Visual Paradigm es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación[16].

1.5.5 Axure RP 10

Es una herramienta muy completa, orientada a diseñar wireframes y prototipos básicos o avanzados de forma fácil, dirigido a aplicaciones web y de escritorio[17].

1.5.6 Visual Studio Code 1.82.3.

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C ++, C #, Java, Python, PHP, Go, .NET) [18].

1.5.7 Frontend

El frontend es la parte del desarrollo web que se dedica a la parte frontal de un sitio web, en pocas palabras del diseño de un sitio web, desde la estructura del sitio hasta los estilos como colores, fondos, tamaños hasta llegar a las animaciones y efectos. Es esa parte de la página con la que interaccionan los usuarios de la misma, es todo el código que se ejecuta en el navegador de un usuario, al que se le denomina una aplicación cliente, es decir, todo lo que el visitante ve y experimenta de forma directa[19].

A continuación, se detallan los lenguajes, frameworks o tecnologías del frontend que son utilizadas en esta investigación.

1.5.7.1 JavaScript.

JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (por ejemplo, programación funcional). Lee más en acerca de JavaScript [20].

1.5.7.2 HTML5.

Es la última versión estable de HTML, HTML5 convierte a HTML de un simple formato de marcado para estructurar documentos en una plataforma completa de desarrollo de aplicaciones. Entre otras características, HTML5 incluye nuevos elementos y API de JavaScript para mejorar el almacenamiento, la multimedia y el acceso al hardware [21].

1.5.7.3 CSS 3.

Hojas de Estilo en Cascada (del inglés Cascading Style Sheets) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (en-US) (incluyendo varios lenguajes basados en XML como SVG, MathML o XHTML). CSS describe como debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios [22].

CSS es uno de los lenguajes base de la Open Web y posee una especificación estandarizada por parte del W3C. Anteriormente, el desarrollo de varias partes de las especificaciones de CSS era realizado de manera sincrónica, lo que permitía el versionado de las recomendaciones [22].

1.5.7.4 Bootstrap 5.3.0.

Sistema de Gestión de Horarios Docentes para la Universidad de Cienfuegos

Bootstrap es un framework CSS y Javascript diseñado para la creación de interfaces limpias y con un diseño responsive. Además, ofrece un amplio abanico de herramientas y funciones, de manera que los usuarios pueden crear prácticamente cualquier tipo de sitio web haciendo uso de los mismos [23].

1.5.8 Backend

El backend procesa la información que alimentará el frontend de datos. Es la capa de acceso a los datos, ya sea de un software o de un dispositivo en general, es la lógica tecnológica que hace que una página web funcione, lo que queda oculto a ojos del visitante[19].

A continuación, se detallan los lenguajes, frameworks o tecnologías del backend.

1.5.8.1 Django 4.1.7.

Django es un framework web de Python de alto nivel que fomenta el desarrollo rápido y el diseño limpio y pragmático. Creado por desarrolladores experimentados, se encarga de gran parte de la molestia del desarrollo web, para que pueda concentrarse en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto [24].

1.5.8.2 Python 3.10.

Python es un lenguaje de programación de código abierto, creado por Guido van Rossum en 1991. Se trata de un lenguaje orientado a objetos, fácil de interpretar y con una sintaxis que permite leerlo de manera semejante a como se lee el inglés. Es un lenguaje interpretado, esto significa que el código de programación se convierte en bytecode y luego se ejecuta por el intérprete, que, en este caso, es la máquina virtual de Python [25].

1.5.9 PostgreSQL 15.

PostgreSQL es un potente sistema de base de datos relacional de objetos de código abierto con más de 35 años de desarrollo activo. Eso le ha valido una sólida reputación de confiabilidad, robustez de funciones y rendimiento [26].

1.6 Conclusiones

Se realizó una descripción del dominio del problema. Se describió el flujo actual de procesos. Se analizaron sistemas existentes vinculados al campo de acción que constituyen buenas herramientas pero que ninguno satisface el problema. Se selecciona la metodología de desarrollo de software AUP para realizar este trabajo y el patrón arquitectónico MTV. Se detallan los lenguajes, frameworks y herramientas utilizadas para el desarrollo de este trabajo.

Capítulo II “Descripción y Construcción de la Solución Propuesta”

2.1 Introducción

Este capítulo está dedicado al proceso de desarrollo de software de la solución propuesta a través de la metodología AUP. Se hace una concepción general del sistema. Se realiza un análisis del proceso de gestión y planificación de horarios para tener una mejor perspectiva de los procesos presentes en el negocio en cuestión. Se extraen y describen los requisitos funcionales y no funcionales del sistema. Se definen las historias de usuarios. Se describen los patrones de diseño. Se definen los patrones de diseño utilizados y el estándar de codificación empleado.

2.2 Concepción general del sistema

La problemática que se aborda en esta tesis se relaciona con las dificultades que enfrenta la Universidad de Cienfuegos en la planificación docente, debido a la falta de herramientas informáticas adecuadas. La solución propuesta implica el desarrollo e implementación de una aplicación web para la gestión de los horarios docentes. Los planificadores van a poder tener control sobre las acciones vinculadas a los horarios docentes de la Universidad de Cienfuegos. El sistema contará con un menú desplegable y dinámico donde se tendrá el control sobre las facultades, carreras, años académicos, grupos y horarios. Los planificadores al confeccionar un horario contarán con los elementos asociados a la confección del mismo como los cursos académicos, los períodos, los días no disponibles, las formas de organización docente, los tipos de actividades extracurriculares, asignaturas, locales y profesores. Una vez que el horario esté confeccionado permitirá exportarlo a formato PDF y JPG ajustándose al modelo del P4. Ver [Anexo 1](#).

Los estudiantes y profesores podrán visualizar y descargar sus horarios a través de la aplicación, lo que les permitirá conocer con anticipación las actividades. Este software permitirá una comunicación fluida con una interfaz amigable, entre los diferentes actores involucrados en la planificación y actualización de los horarios.

2.3 Requisitos de software

Los requisitos de software son la descripción de las características y las funcionalidades del sistema. Los requisitos nos comunican las expectativas de los consumidores de productos de software. Los requisitos pueden ser obvios o estar ocultos, conocidos o desconocidos, esperados o inesperados, desde el punto de vista del cliente [27].

2.3.1 Requisitos Funcionales

A partir de entrevistas realizadas a los planificadores, se obtuvieron 68 requisitos funcionales (RF) los cuales se muestran en la siguiente tabla.

Nº	Nombre	Descripción	Prioridad	Complejidad
RF1	Autenticar usuario	El usuario introduce sus datos, el sistema lo comprueba y si son válidos, el usuario queda autenticado con el nivel de privilegios asignados; si los datos no son válidos el sistema muestra un mensaje de error.	Alta	Alta
RF2	Salir del sistema	El usuario sale del sistema y lo lleva a la página de inicio de sesión.	Alta	Baja
RF3	Crear Curso	El sistema permite que los usuarios con rol planificador puedan crear un curso de clases.	Alta	Baja

Sistema de Gestión de Horarios Docentes para la Universidad de Cienfuegos

Nº	Nombre	Descripción	Prioridad	Complejidad
RF4	Mostrar Curso	El sistema permite mostrar todos los cursos a los usuarios con rol planificador y solo los cursos visibles a los usuarios con los roles de estudiante y profesor.	Alta	Media
RF5	Editar Curso	El sistema permite que los usuarios con rol planificador puedan editar un curso de clases ya existente.	Alta	Baja
RF6	Eliminar Curso	El sistema permite que los usuarios con rol planificador puedan eliminar un curso de clases ya existente.	Alta	Media
RF7	Crear Período	El sistema permite que los usuarios con rol planificador puedan crear un período de un curso de clases.	Alta	Baja
RF8	Mostrar Período	El sistema permite mostrar todos los períodos de los cursos a los usuarios con rol planificador y solo los períodos visibles a los usuarios con los roles de estudiante y profesor.	Alta	Media
RF9	Editar Período	El sistema permite que los usuarios con rol planificador puedan editar un período de un curso de clases ya existente.	Alta	Baja
RF10	Eliminar Período	El sistema permite que los usuarios con rol planificador puedan eliminar un período de un curso de clases ya existente.	Alta	Media
RF11	Crear Facultad	El sistema permite que los usuarios con rol planificador puedan crear una facultad.	Alta	Baja
RF12	Mostrar Facultad	El sistema permite mostrar a todos los usuarios todas las facultades.	Alta	Baja
RF13	Editar Facultad	El sistema permite que los usuarios con rol planificador puedan editar una facultad ya existente.	Alta	Baja
RF14	Eliminar Facultad	El sistema permite que los usuarios con rol planificador puedan eliminar una facultad ya existente.	Alta	Media
RF15	Crear Carrera	El sistema permite que los usuarios con rol planificador puedan crear una carrera de una facultad.	Alta	Baja
RF16	Mostrar Carrera	El sistema permite mostrar a todos los usuarios todas las carreras de las diferentes facultades.	Alta	Baja
RF17	Editar Carrera	El sistema permite que los usuarios con rol planificador puedan editar una carrera ya existente.	Alta	Baja
RF18	Eliminar Carrera	El sistema permite que los usuarios con rol planificador puedan eliminar una carrera ya existente.	Alta	Media

Sistema de Gestión de Horarios Docentes para la Universidad de Cienfuegos

Nº	Nombre	Descripción	Prioridad	Complejidad
RF19	Crear Año	El sistema permite que los usuarios con rol planificador puedan crear un año de una carrera.	Alta	Baja
RF20	Mostrar Año	El sistema permite mostrar a todos los usuarios todos los años de las diferentes carreras.	Alta	Baja
RF21	Eliminar Año	El sistema permite que los usuarios con rol planificador puedan eliminar un año ya existente.	Alta	Media
RF22	Crear Grupo	El sistema permite que los usuarios con rol planificador puedan crear un grupo de un año.	Alta	Baja
RF23	Mostrar Grupo	El sistema permite mostrar a todos los usuarios todos los grupos de los diferentes años.	Alta	Baja
RF24	Editar Grupo	El sistema permite que los usuarios con rol planificador puedan editar un grupo ya existente.	Alta	Baja
RF25	Eliminar Grupo	El sistema permite que los usuarios con rol planificador puedan eliminar un grupo ya existente.	Alta	Media
RF26	Crear Asignatura	El sistema permite que los usuarios con rol planificador puedan crear una asignatura.	Alta	Baja
RF27	Mostrar Asignatura	El sistema permite mostrar a los usuarios con rol planificador todas las asignaturas.	Alta	Baja
RF28	Editar Asignatura	El sistema permite que los usuarios con rol planificador puedan editar una asignatura ya existente.	Alta	Baja
RF29	Eliminar Asignatura	El sistema permite que los usuarios con rol planificador puedan eliminar una asignatura ya existente.	Alta	Baja
RF30	Agregar días no disponibles	El sistema permite que los usuarios con rol planificador puedan agregar días no disponibles a un período.	Media	Baja
RF31	Mostrar días no disponibles	El sistema permite mostrar a todos los usuarios todos los días no disponibles de un período en la vista de horario.	Alta	Baja
RF32	Editar días no disponibles	El sistema permite que los usuarios con rol planificador puedan editar los días no disponibles de un período.	Media	Baja
RF33	Eliminar días no disponibles	El sistema permite que los usuarios con rol planificador puedan eliminar días no disponibles de un período.	Media	Baja
RF34	Agregar local	El sistema permite que los usuarios con rol planificador puedan agregar un local.	Alta	Baja

Sistema de Gestión de Horarios Docentes para la Universidad de Cienfuegos

Nº	Nombre	Descripción	Prioridad	Complejidad
RF35	Mostrar local	El sistema permite mostrar a todos los usuarios el local de un grupo en un período determinado y de cada turno de clases en la vista de horario.	Alta	Baja
RF36	Editar local	El sistema permite que los usuarios con rol planificador puedan editar un local.	Alta	Baja
RF37	Eliminar local	El sistema permite que los usuarios con rol planificador puedan eliminar un local.	Alta	Baja
RF38	Agregar turno de clases	El sistema permite que los usuarios con rol planificador puedan agregar un turno de clases en la vista de editar horario.	Alta	Alta
RF39	Mostrar turnos de clases	El sistema permite mostrar a todos los usuarios todos los turnos de clases de un grupo en un período determinado en la vista de horario.	Alta	Alta
RF40	Editar turno de clases	El sistema permite que los usuarios con rol planificador puedan editar los turnos de clases desde la vista editar horario.	Alta	Alta
RF41	Eliminar turnos de clases	El sistema permite que los usuarios con rol planificador puedan eliminar turnos de clases desde la vista editar horario.	Alta	Alta
RF42	Agregar actividad	El sistema permite que los usuarios con rol planificador puedan agregar una actividad.	Media	Baja
RF43	Mostrar actividad	El sistema permite mostrar a todos los usuarios todas las actividades en la vista de horario.	Media	Baja
RF44	Editar actividad	El sistema permite que los usuarios con rol planificador puedan editar las actividades.	Media	Baja
RF45	Eliminar actividades	El sistema permite que los usuarios con rol planificador puedan eliminar una actividad.	Media	Baja
RF46	Crear horario	El sistema permite que los usuarios con rol planificador puedan crear un horario.	Alta	Baja
RF47	Mostrar horario	El sistema permite mostrar a todos los usuarios todos los horarios.	Alta	Alta
RF48	Editar datos del horario	El sistema permite que los usuarios con rol planificador puedan editar los datos del horario.	Alta	Alta
RF49	Editar horario	El sistema permite que los usuarios con rol planificador puedan editar los horarios.	Alta	Alta
RF50	Eliminar horario	El sistema permite que los usuarios con rol planificador puedan eliminar un horario.	Alta	Media
RF51	Agregar semanas no disponibles	El sistema permite que los usuarios con rol planificador puedan agregar	Media	Media

Sistema de Gestión de Horarios Docentes para la Universidad de Cienfuegos

Nº	Nombre	Descripción	Prioridad	Complejidad
		semanas no disponibles a un horario.		
RF52	Mostrar semanas no disponibles	El sistema permite mostrar a todos los usuarios todos las semanas no disponibles en la vista de horario.	Media	Media
RF53	Editar semanas no disponibles	El sistema permite que los usuarios con rol planificador puedan editar las semanas no disponibles de un horario.	Media	Baja
RF54	Cambiar a semana disponible	El sistema permite que los usuarios con rol planificador puedan cambiar a semana disponible una semana no disponible de un horario.	Media	Media
RF55	Exportar P4 a pdf	El sistema permite que todos los usuarios puedan exportar el P4 a formato pdf.	Alta	Media
RF56	Exportar P4 a jpg	El sistema permite que todos los usuarios puedan exportar el P4 a formato jpg.	Alta	Media
RF57	Agregar actividades extracurriculares	El sistema permite que los usuarios con rol planificador puedan agregar actividades extracurriculares a un horario.	Media	Alta
RF58	Mostrar actividades extracurriculares	El sistema permite mostrar a todos los usuarios los turnos donde hay actividades extracurriculares en la vista de horario.	Media	Alta
RF59	Editar actividades extracurriculares	El sistema permite que los usuarios con rol planificador puedan editar actividades extracurriculares del horario.	Media	Alta
RF60	Eliminar actividades extracurriculares	El sistema permite que los usuarios con rol planificador puedan eliminar actividades extracurriculares del horario.	Media	Alta
RF61	Agregar tipos de actividades extracurriculares	El sistema permite que los usuarios con rol planificador puedan agregar tipos de actividades extracurriculares.	Media	Alta
RF62	Mostrar tipos de actividades extracurriculares	El sistema permite mostrar a todos los usuarios todos los tipos de actividades extracurriculares en la vista de horario.	Media	Alta
RF63	Editar tipos de actividades extracurriculares	El sistema permite que los usuarios con rol planificador puedan editar tipos de actividades extracurriculares.	Media	Alta
RF64	Eliminar tipos de actividades extracurriculares	El sistema permite que los usuarios con rol planificador puedan eliminar tipos de actividades extracurriculares del horario.	Media	Alta
RF65	Agregar profesor	El sistema permite que los usuarios con rol planificador puedan agregar un local.	Alta	Baja
RF66	Mostrar profesores	El sistema permite mostrar a todos los usuarios los profesores de las	Alta	Media

Sistema de Gestión de Horarios Docentes para la Universidad de Cienfuegos

Nº	Nombre	Descripción	Prioridad	Complejidad
		asignaturas de los turnos de clases en la vista del horario.		
RF67	Editar profesor	El sistema permite que los usuarios con rol planificador puedan editar un profesor.	Alta	Media
RF68	Eliminar profesor	El sistema permite que los usuarios con rol planificador puedan eliminar un profesor.	Alta	Baja

Tabla 1 Descripción de los requisitos funcionales del sistema.

2.3.2 Requisitos no funcionales

Los requisitos no funcionales (RNF) se refieren a las características o cualidades que debe tener el producto para hacerlo atractivo, usable, rápido y confiable. Para el desarrollo del sistema se definieron 5 RNF, los que se clasificaron en RNF de Interfaz o apariencia externa, de software, de hardware, de seguridad y privacidad, de rendimiento y de mantenibilidad, que se encuentran a continuación:

Nº	Tipo de requisito	Descripción
RNF1	Interfaz o apariencia externa	La interfaz debe ser sencilla, intuitiva, legible, amigable y mantener el formato en vistas similares.
RNF2	Software	Se requiere de un navegador web instalado en las computadoras clientes.
RNF3	Seguridad y privacidad	La información estará protegida contra accesos no autorizados mediante mecanismos de autenticación y encriptación.
RNF4	Rendimiento	La aplicación debe cumplir con los requisitos funcionales en el menor tiempo posible sin sacrificar Mantenibilidad.
RNF5	Mantenibilidad	Lograr una aplicación que a medida que vaya creciendo en tamaño no lo haga en complejidad, con el uso de buenas prácticas de programación, y por lo tanto sea más fácil el desarrollo de nuevas versiones de la misma.

Tabla 2 Descripción de los requisitos no funcionales del sistema.

2.3.3 Historias de usuario

Las historias de usuarios son las técnicas utilizadas para especificar los requisitos del software, son equivalentes a los casos de uso en el proceso unificado y constituyen la base para las pruebas funcionales [28].

En total se definieron 68 historias de usuario, a continuación, se muestra la historia de usuario Mostrar Horario.

Número: 47	Requisito: Mostrar Horario
Programador: Leonardo Alain Moreira Rodríguez	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 20 horas
Riesgo en desarrollo:	Tiempo Real: 72 horas
Descripción: El componente permitirá mostrar todas las semanas de un horario a los usuarios del sistema. PDF: Al presionar en este botón se generará un archivo pdf del P4 de este horario. JPG: Al presionar en este botón se generará un archivo jpg del P4 de este horario. Editar: Al presionar este botón llevará al usuario con rol planificador a la vista de editar el horario	
Observaciones:	

Prototipo elemental de interfaz gráfica de usuario:

	PDF	JPG	Editar	Tema
--	-----	-----	--------	------

Semana 1/8
04/09/23 - 09/09/23

	Lun	Mar	Mié	Jue	Vie	Sáb
	4	5	6	7	8	9
8:30 am	M	EE	S	B	B	
9:30 am	M	B	B	B	B	
10:35 am	S	B	B	M	EE	
11:40 am						
12:45 pm	S	EF	E	M	S	
13:50 pm						
1:50 pm						
2:55 pm						

Semana 2/8
11/09/23 - 16/09/23

	Lun	Mar	Mié	Jue	Vie	Sáb
	11	12	13	14	15	16
8:30 am						
9:30 am						
10:35 am						
11:40 am						
12:45 pm						
13:50 pm						
1:50 pm						
2:55 pm						

Semana 3/8
18/09/23 - 23/09/23

	Lun	Mar	Mié	Jue	Vie	Sáb
	18	19	20	21	22	23
8:30 am	EE	B	B	B	B	
9:30 am	EE	B	B	B	B	
10:35 am	S	E	M	E	S	
11:40 am						
12:45 pm	S	E	M	E	S	
13:50 pm						
1:50 pm						
2:55 pm						

Semana 4/8
25/09/23 - 30/09/23

	Lun	Mar	Mié	Jue	Vie	Sáb
	25	26	27	28	29	30
8:30 am						
9:30 am						
10:35 am						
11:40 am						
12:45 pm						
13:50 pm						
1:50 pm						
2:55 pm						

Semana 5/8
02/10/23 - 07/10/23

	Lun	Mar	Mié	Jue	Vie	Sáb
	2	3	4	5	6	7
8:30 am	M		B	B		
9:30 am	M		B	B		
10:35 am	S		M	EE		
11:40 am						
12:45 pm	S		M	EE		
13:50 pm						
1:50 pm						
2:55 pm						

Semana 6/8
09/10/23 - 14/10/23

	Lun	Mar	Mié	Jue	Vie	Sáb
	9	10	11	12	13	14
8:30 am						
9:30 am						
10:35 am						
11:40 am						
12:45 pm						
13:50 pm						
1:50 pm						
2:55 pm						

Semana 7/8
16/10/23 - 21/10/23

	Lun	Mar	Mié	Jue	Vie	Sáb
	16	17	18	19	20	21
8:30 am	B	B	M	S	B	
9:30 am	B	B	M	S	B	
10:35 am	S	E	S	M	EE	
11:40 am						
12:45 pm	S	E	S	M	EE	
13:50 pm						
1:50 pm						
2:55 pm						

Semana 8/8
23/10/23 - 28/10/23

	Lun	Mar	Mié	Jue	Vie	Sáb
	23	24	25	26	27	28
8:30 am						
9:30 am						
10:35 am						
11:40 am						
12:45 pm						
13:50 pm						
1:50 pm						
2:55 pm						

Tabla 3 Historia de Usuario del requisito Funcional Mostrar Horario

2.4 Descripción de la arquitectura

La arquitectura del software constituye una especificación de las principales ideas del diseño proporcionando una descripción más detallada de cómo realizar dicho sistema[29].

Patrón de la arquitectura

La propuesta de solución se basa en el patrón arquitectónico Modelo Template Vista.

Modelo: Maneja todo lo relacionado con la información, esto incluye como acceder a esta, la validación, relación entre los datos y su comportamiento.

Vista: Es un enlace entre el modelo y el template. Decide qué información será mostrada y por cual template.

Template: Decide como será mostrada la información.

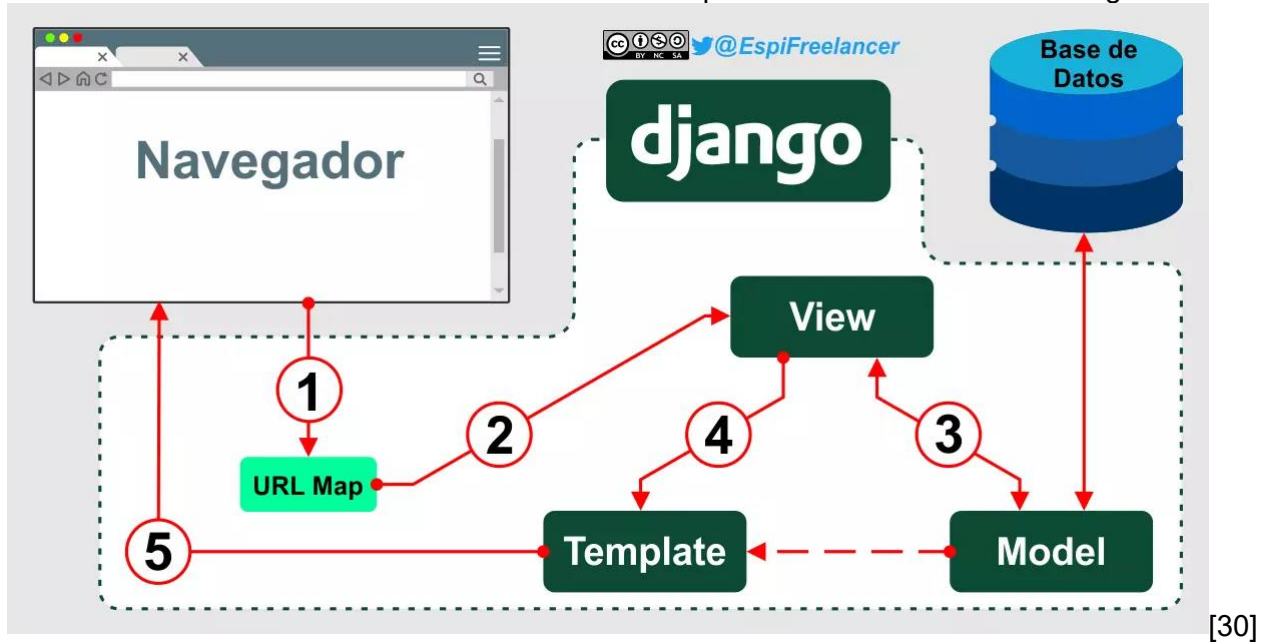


Figura 1 Arquitectura Modelo Template Vista

Al momento de hacer click en un enlace o escribir una dirección (1) a lo primero que se accede es al mapa de URLs (también conocido como URL map o URL conf), en este archivo cada ruta está asociada con una view (2), si se necesita algún dato se solicitará este a model (3), el cual a su vez generará la consulta a la base de datos, cuando los datos han sido traídos estos son enviados al template (4) que contiene la lógica de presentación para estos. Luego de "pintar" la página esta se envía al navegador que hizo la solicitud (5)[31].

A continuación, se muestra la aplicación de la arquitectura a la propuesta de solución:

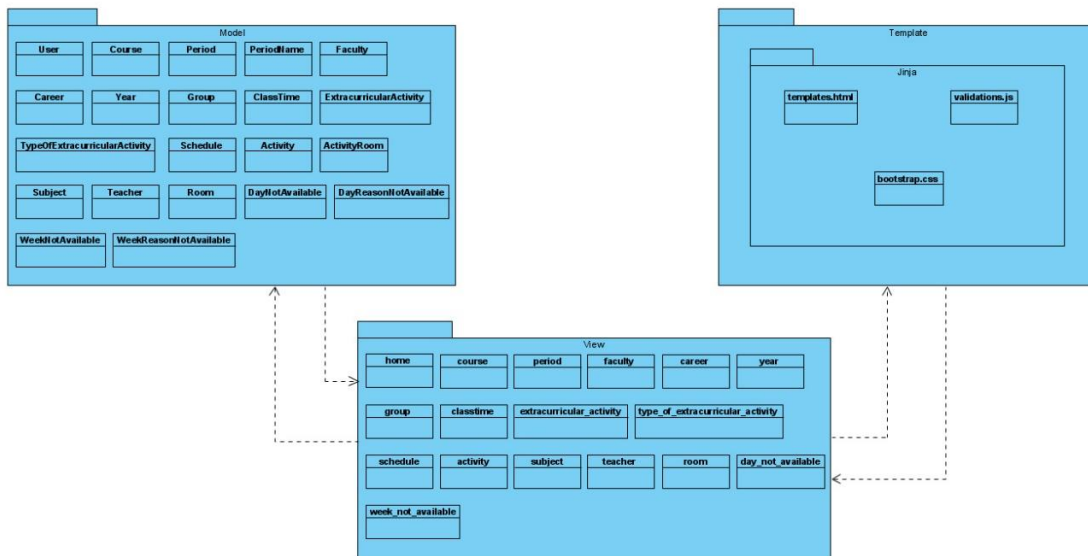


Figura 2 Aplicación de la arquitectura a la solución propuesta.

2.5 Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases. Tiene el propósito de describir las clases que conforman el modelo de un determinado sistema. Dado el carácter de refinamiento iterativo que caracteriza un desarrollo

Sistema de Gestión de Horarios Docentes para la Universidad de Cienfuegos orientado a objetos, el diagrama de clase va a ser creado y refinado durante las fases de análisis y diseño, estando presente como guía en la implementación del sistema[29].

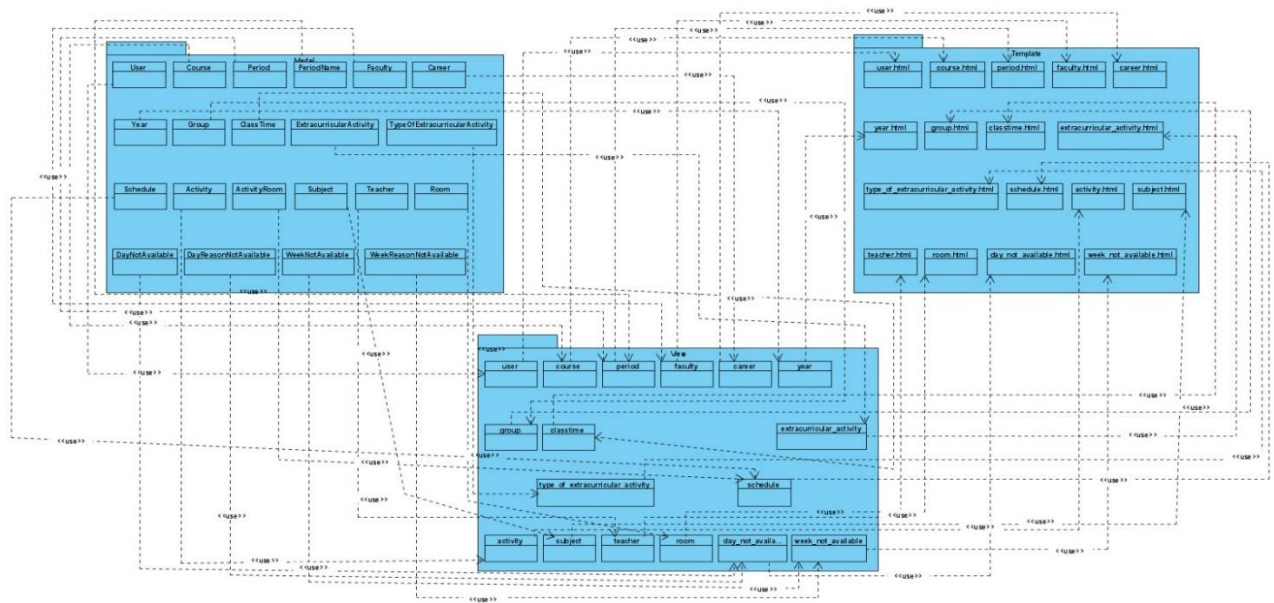


Figura 3 Diagrama de clases del diseño.

2.6 Modelo de la base de datos

Una base de datos se puede definir como: Una colección o depósito de datos integrados con redundancia controlada y con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real. Los datos que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de estas, y su definición y descripción únicas para cada tipo de datos, han de estar almacenadas juntos con los mismos. Los procedimientos de actualización y recuperación, comunes y bien determinados, habrán de ser capaces de conservar la seguridad (integridad, confidencialidad y disponibilidad) del conjunto de los datos[32].

El modelo de Base de Datos es el conjunto de componentes o herramientas conceptuales que un Sistema de Gestión de Bases de Datos proporciona para modelar[33].

Sistema de Gestión de Horarios Docentes para la Universidad de Cienfuegos

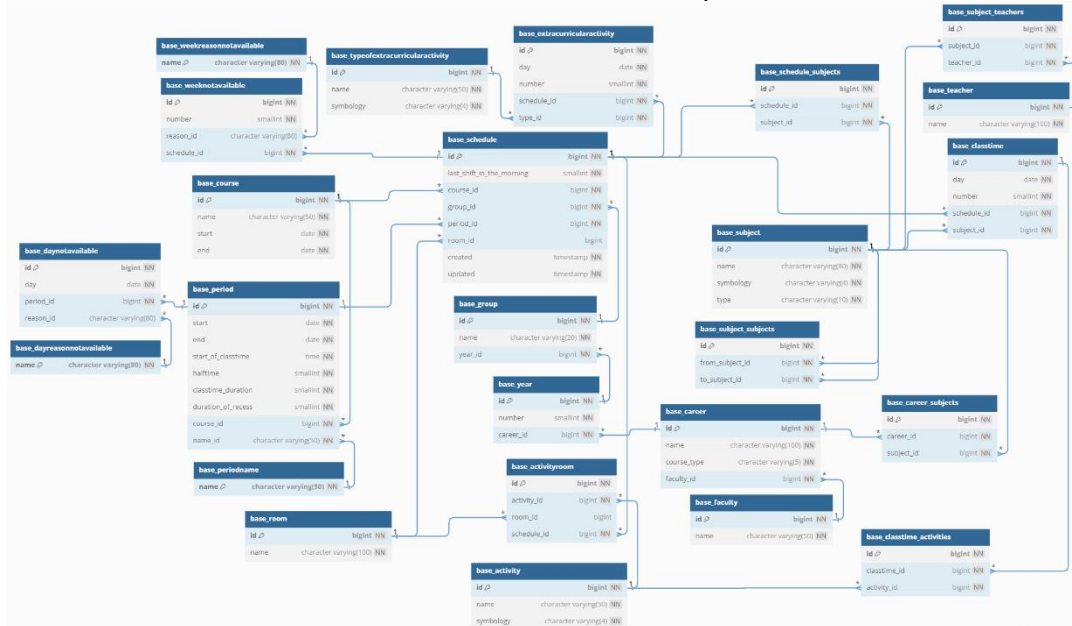


Figura 4 Modelo de datos del sistema.

2.7 Patrones de diseño utilizados

Experto: es el principio básico de asignación de responsabilidades. A una clase le corresponde la responsabilidad de lo referente a un objeto.

Creador: asigna responsabilidades de creación de un objeto a la clase que contiene la información necesaria para crearlo.

Alta cohesión: expresa que la información que almacena una clase debe ser coherente y está en la mayor medida de lo posible relacionada con la clase.

Bajo acoplamiento: define que las clases deben depender lo menos posible una de otra, para que las modificaciones resulten simples y limpias.

2.8 Estándar de codificación empleado

Con el objetivo de lograr una estandarización en la programación del sistema se decide aplicar el estándar de codificación snake_case para nombres de variables y funciones en Python, se utiliza UpperCamelCase para el nombre de las clases en Python y Javascript, se utiliza lowerCamelCase para el nombre de las variables y funciones en Javascript, se utiliza SCREAMING_SNAKE_CASE para el nombre de las constantes en Python, se utiliza kebab-case para el nombre de las clases en HTML. A continuación, se describe a grandes rasgos las convenciones de nomenclatura.

Convenciones de nomenclatura

General

- Todo el código será programado en inglés, exceptuando los casos en que por determinados motivos no tiene una traducción específica, para evitar el uso de caracteres extraños como la ñ y las tildes, y lograr inteligibilidad y uniformidad en el código.
- Se utilizarán nombres que sean claros, concretos y libres de ambigüedades. Ejemplo: "course_id" y no solamente "id".
- El nombre de todas las variables y métodos comenzarán con letra minúscula y si este está compuesto por varias palabras, todas las palabras internas que lo componen estarán separadas por un guion bajo. Ejemplo: "delete_period".

Clases

- Intentar mantener los nombres de las clases descriptivos y simples. Usar palabras completas, evitar acrónimos y abreviaturas, a no ser que la abreviatura sea mucho más conocida que el nombre completo, como URL o HTML.

Nombre de variables

- No se utilizarán nombres de variables que puedan ser ambiguos.
- Se procurará evitar dar nombres sin sentido a variables temporales. Por ejemplo: temp, i, tmp.
- Las variables booleanas deben tener nombres que sugieran respuestas o contenidos de tipo Sí/No, por ejemplo: "created".
- Los nombres de las variables booleanas deben ser positivos, por ejemplo: "is/true".

2.9 Conclusiones

Se hizo una concepción general del sistema. Se realizó un análisis del proceso de gestión de horarios docentes. Se extrajeron 59 requisitos funcionales y 5 requisitos no funcionales. Se hizo una descripción detallada de las historias de usuario con sus prototipos. Se describió el patrón de arquitectura empleado. Se mostró la aplicación de la arquitectura a la propuesta de solución. Se diseñó el diagrama de clases del sistema. Se diseñaron los modelos de la base de datos. Se definieron los patrones de diseño utilizados y el estándar de codificación empleado.

Capítulo III “Validación de la Solución Propuesta”

3.1 Introducción

Este capítulo está dedicado al proceso de validación de la solución propuesta. Se aplicaron métricas con el objetivo de medir la calidad de la especificación de los requisitos. Se realiza el diseño de las pruebas funcionales y de aceptación para encontrar posibles errores en el sistema.

3.2 Métrica aplicada a los requerimientos

Con el objetivo de medir la calidad de la especificación de los requisitos se aplicó una de las métricas Calidad de la especificación (CE). Para obtener cuán entendibles y precisos son los requisitos, primeramente, se calcula el total de requisitos de la especificación como se muestra a continuación:

Nr: El total de requisitos de especificación.

Nf: Cantidad de requisitos funcionales.

Nnf: Cantidad de requisitos no funcionales.

Como resultado de la sustitución de los valores, para el sistema se obtiene:

$$Nr = Nf + Nnf$$

$$Nr = 68 + 5$$

$$Nr = 73$$

Para determinar, finalmente, la Especificidad de los Requisitos (ER) o ausencia de ambigüedad en los mismos se realiza la siguiente operación:

$$ER = Nui / Nr$$

Donde Nui es el número de requisitos para los cuales todos los revisores tuvieron interpretaciones idénticas. Mientras más cerca de 1 esté el valor de ER, menor será la ambigüedad.

Para el caso de los requisitos obtenidos solo uno produjo contradicción en las interpretaciones. Sustituyendo las variables se obtiene:

$$ER = 72 / 73$$

$$ER = 0.98$$

Arrojando un resultado final satisfactorio, indicando que el grado de ambigüedad de los requisitos es sumamente bajo (2%) ya que el 98% es entendible. Los requisitos ambiguos fueron modificados y validados para garantizar su correcta comprensión.

3.3 Pruebas de caja blanca

La prueba de caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo; ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa; ejecuten todos los bucles en sus límites y con sus límites operacionales; y ejerciten las estructuras internas de datos para asegurar su validez[28].

Para comprobar que las funciones internas del sistema se ejecutan correctamente se le realizaron pruebas al código de las principales funcionalidades del sistema. Para ello se empleó la técnica de la **prueba del camino básico**, el cual consiste en garantizar que se pueda probar al menos una

Sistema de Gestión de Horarios Docentes para la Universidad de Cienfuegos
vez cada una de las sentencias del programa, partiendo de la obtención de la medida de la complejidad de un procedimiento o algoritmo y la obtención de un conjunto básico de caminos de ejecución de este, los cuales son utilizados para obtener los casos de prueba[28].

A continuación, se muestra el proceso de pruebas realizado a la vista `create_faculty`. Primeramente, se comienza por analizar el código y enumerar las instrucciones:

```
@login_required(login_url='authentication:login')
@allowed_users(allowed_roles=['Planificador'])
def create_faculty(request):
    form = FacultyForm() # 1
    if request.method == 'POST': # 2
        accept = request.POST.get('_acceptar') # 3
        form = FacultyForm(request.POST) # 4
        if form.is_valid(): # 5
            form.save() # 6
            if accept: # 7
                return redirect('base:home') # 8
            return redirect('base:create_faculty') # 9

    context = {'form': form} # 10
    return render(request, 'base/create_faculty.html', context=context) # 11
```

Figura 5 Vista `create_faculty` para agregar una Facultad.

Para obtener los casos de prueba a partir de la técnica seleccionada se debe construir seguidamente el grafo de flujo correspondiente al código de la función como se muestra en la Figura.

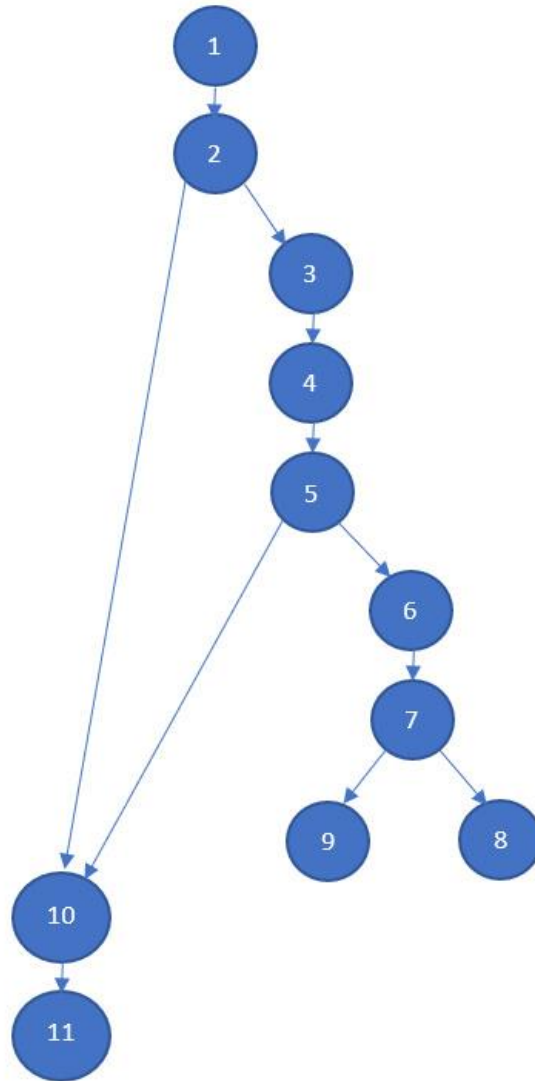


Figura 6 Grafo de flujo del código de la vista create_faculty

Luego se determina la complejidad ciclomática $V(G)$ del grafo resultante, la cual es un indicador del número de caminos independientes que existen en un grafo, es decir, es cualquier camino dentro del código que introduce por lo menos un nuevo conjunto de sentencias de proceso o una nueva condición. La complejidad ciclomática puede ser calculada de la siguiente forma:

$V(G) = c + 1$, siendo c el número de nodos de condición

Realizando los cálculos correspondientes se obtiene el siguiente resultado:

$$V(G) = c + 1$$

$$V(G) = 4$$

Por lo que el conjunto de caminos básico sería:

Camino básico 1: 1-2-3-4-5-6-7-8

Camino básico 2: 1-2-3-4-5-6-7-9

Camino básico 3: 1-2-3-4-5-10-11

Luego se definen los casos de prueba para cada uno de los caminos básicos obtenidos. A continuación, se muestra el resultado de las pruebas aplicadas a todos los caminos básicos.

Descripción: se verificará que se cree una nueva facultad		
Condición de ejecución:	Entrada:	Resultados esperados
El método de la petición request debe ser POST, haber llenado correctamente los campos del formulario y se debe haber presionado en el botón Aceptar del formulario.	Formulario llenado correctamente	Se agrega correctamente la nueva facultad y navega a la vista home del sitio.
Resultado obtenido: Satisfactorio.		

Tabla 4 Caso de prueba del camino básico 1

Descripción: se verificará que se cree una nueva facultad y permita agregar otra		
Condición de ejecución:	Entrada:	Resultados esperados
El método de la petición request debe ser POST, haber llenado correctamente los campos del formulario y se debe haber presionado en el botón Aceptar y seguir agregando del formulario.	Formulario llenado correctamente	Se agrega correctamente la nueva facultad y redirecciona a la misma vista create_faculty para agregar otra facultad.
Resultado obtenido: Satisfactorio.		

Tabla 5 Caso de prueba del camino básico 2

Descripción: se verificará que no se cree una nueva facultad		
Condición de ejecución:	Entrada:	Resultados esperados
El método de la petición request debe ser POST, haber llenado los campos del formulario y se debe haber presionado en el botón Aceptar del formulario.	Formulario llenado incorrectamente	No se agrega la facultad y se muestra los respectivos mensajes de error.
Resultado obtenido: Satisfactorio.		

Tabla 6 Caso de prueba del camino básico 3

Descripción: se verificará que se muestre bien el formulario		
Condición de ejecución:	Entrada:	Resultados esperados
El método de la petición request debe ser GET, se debe haber presionado en el botón agregar facultad.	El formulario debe mostrarse vacío.	El formulario se muestra vacío para permitir al usuario agregar una facultad.
Resultado obtenido: Satisfactorio.		

Se logró el cálculo de la complejidad ciclomática para los algoritmos de la aplicación. La misma visualiza la forma que se emplea para obtener dicho cálculo con las estructuras especificadas. Los resultados obtenidos en el cálculo de la complejidad ciclomática definen el número de caminos independientes dentro de un fragmento de código y la cantidad de casos de pruebas diseñados.

3.4 Pruebas de caja negra

Las pruebas de caja negra, también denominada “prueba de comportamiento”, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los detectados por los métodos de caja blanca. La prueba de caja negra intenta encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación[29].

Para desarrollar las pruebas de caja negra se utilizó la técnica, de la Partición de Equivalencia; esta es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores[28].

A continuación, se muestra el proceso de pruebas de partición de equivalencia realizado al Requisito Funcional **Crear Período**.

Descripción

Los datos de entrada y los resultados de salida se agrupan en clases diferentes, en las que todos los miembros de dicha clase están relacionados[34].

Cada una de estas clases es una partición de equivalencia en la que el programa se comporta de la misma forma para cada miembro de la clase[34].

Se supone que la prueba de un valor representativo de cada clase sea equivalente a la prueba de cualquier otro valor[34].

Identificación de las clases de equivalencia

Se toma cada condición de entrada o salida (usualmente frase o sentencia en la especificación) y se la divide en 1 o más grupos. Se identifican dos tipos de clase de equivalencias: validas e invalidas[34].

Una cierta combinación de clases de entrada daría como resultado una combinación de clases de salida[34].

A continuación, se muestran las clases de equivalencia para los campos de entrada del formulario del Requisito Funcional **Crear Período**.

Campo: Nombre	
Tipo: Texto	
Clases válidas	Clases no válidas
Menos de 50 caracteres	Más de 50 caracteres

Tabla 8 Clases de equivalencia para el Campo Nombre

Campos: Inicio y Fin	
Tipo: Fecha	
Clases válidas	Clases no válidas
Mayor o igual que la fecha de inicio del curso al cual se está agregando	Menor que la fecha de inicio del curso al cual se está agregando
Menor o igual que la fecha de fin del curso al cual se está agregando	Mayor que la fecha de fin del curso al cual se está agregando
Día de la semana cualquiera menos domingo	Día de la semana domingo
Campo de Inicio menor que el de Fin	Campo de Inicio mayor que el de Fin
Cantidad de semanas comprendidas entre el rango de fechas del Campo de Inicio y del Campo de Fin es mayor o igual que 3	Cantidad de semanas comprendidas entre el rango de fechas del Campo de Inicio y del Campo de Fin es menor que 3
Cantidad de semanas comprendidas entre el rango de fechas del Campo de Inicio y del Campo de Fin es menor o igual que 24	Cantidad de semanas comprendidas entre el rango de fechas del Campo de Inicio y del Campo de Fin es mayor que 24

Tabla 9 Clases de equivalencia para los Campos Inicio y Fin

Campos: Inicio del turno de clases, Entretiempo, Duración del turno de clases, Duración del receso	
Tipos: Hora, Campo de selección	
Clases válidas	Clases no válidas
El cálculo de las horas totales de duración de clases de un día a través de estos campos, contadas a partir de las 12:00 AM, es menor o igual que 23.99	El cálculo de las horas totales de duración de clases de un día a través de estos, contadas a partir de las 12:00 AM, campos es mayor que 23.99

Tabla 10 Clases de equivalencia para los Campos Inicio del turno de clases, Entretiempo, Duración del turno de clases y Duración del receso

Definición de los casos de prueba

Por cada dato de entrada y salida se elige un valor perteneciente a una clase de equivalencia. Un caso de prueba está formado por un dato de prueba para cada dato de entrada y salida (resultado esperado)[34].

Pautas:

- El objetivo es probar todas las clases válidas y no válidas al menos una vez cada una de ellas [34].
- Cada caso debe cubrir el máximo número de entradas [34].
- En un caso de prueba pueden aparecer varias clases de entrada válidas [34].
- En un caso de prueba solamente puede aparecer una entrada no válida. El criterio para las clases no válidas impide que los errores se enmascaren mutuamente [34].
- Se debe especificar el resultado esperado [34].

Diseño de los casos de prueba

Se denotó cada caso de prueba con letras mayúsculas en orden alfabético de la A a la J para un total de 10 casos de prueba. Por cada caso de prueba se realizó una prueba unitaria. Ver [Anexo 2](#).

Caso de prueba: A			
Valores de entrada	Valores salida(esperados)	de	Valores obtenidos
Todos los campos válidos	True		True
Agregar Período al Curso 20 </			

Tabla 11 Caso de prueba A de particiones de equivalencia

Caso de prueba: B			
Valores de entrada		Valores de salida(esperados)	Valores obtenidos
Campo Nombre no válido por tener más de 50 caracteres. Los restantes campos válidos		False	False

Tabla 12 Caso de prueba B de particiones de equivalencia

32

Agregar Período al Curso 20

×

Nombre

Semestre I

La fecha de inicio del período tiene que estar en el rango del Curso 20 del 06/01/2020 al 19/12/2020

Inicio

13/01/2019

Fin

27/06/2020

Inicio de turno de clases

08:00 AM

Entretiempo

10 minutos

Duración del turno de clases

90 minutos

Duración del receso

0 minutos

Aceptar

Resultado obtenido: Satisfactorio

Tabla 13 Caso de prueba C de particiones de equivalencia

Caso de prueba: D		
Valores de entrada	Valores salida(esperados)	de Valores obtenidos
Campo Inicio no válido por ser una fecha con día de la semana domingo. Los restantes campos válidos	False	False

✕

La fecha de inicio no puede ser domingo

Nombre

Semestre I
 ▼

Inicio

19/01/2020
 📅

Fin

27/06/2020
 📅

Inicio de turno de clases

08:00 AM
 🕒

Entretiempo

10 minutos
 ▼

Duración del turno de clases

90 minutos
 ▼

Duración del receso

0 minutos
 ▼

Aceptar

Resultado obtenido: Satisfactorio

Tabla 14 Caso de prueba D de particiones de equivalencia

Caso de prueba: E		
Valores de entrada	Valores salida(esperados)	de Valores obtenidos
Campo Fin no válido por estar fuera del rango de fecha del curso. Los restantes campos válidos	False	False

Agregar Período al Curso 20

×

Nombre

Semestre I

▼

Inicio

07/09/2020

📅

La fecha de fin del período tiene que estar en el rango del Curso 20 del 06/01/2020 al 19/12/2020

Fin

09/01/2021

📅

Inicio de turno de clases

08:00 AM

🕒

Entretiempo

10 minutos

▼

Duración del turno de clases

90 minutos

▼

Duración del receso

0 minutos

▼

Aceptar

Resultado obtenido: Satisfactorio

Tabla 15 Caso de prueba E de particiones de equivalencia

Caso de prueba: F		
Valores de entrada	Valores salida(esperados)	de Valores obtenidos
Campo Fin no válido por ser una fecha con día de la semana domingo. Los restantes campos válidos	False	False

Agregar Período al Curso 20

La fecha de fin no puede ser domingo

Nombre

Semestre I

Inicio

13/01/2020

Fin

21/06/2020

Inicio de turno de clases

08:00 AM

Entretiempo

10 minutos

Duración del turno de clases

90 minutos

Duración del receso

0 minutos

Aceptar

Resultado obtenido: Satisfactorio

Tabla 16 Caso de prueba F de particiones de equivalencia

Caso de prueba: G		
Valores de entrada	Valores de salida(esperados)	Valores obtenidos
Campo Inicio no es válido por ser mayor que el campo Fin. Los restantes campos válidos	False	False

Agregar Período al Curso 20
✕

La fecha de inicio no puede ser mayor que la fecha de fin

Nombre

Semestre I
▼

Inicio

13/04/2020

📅

Fin

13/01/2020

📅

Inicio de turno de clases

08:00 AM

🕒

Entretiempo

10 minutos
▼

Duración del turno de clases

90 minutos
▼

Duración del receso

0 minutos
▼

Aceptar

Resultado obtenido: Satisfactorio

Tabla 17 Caso de prueba G de particiones de equivalencia

Caso de prueba: H		
Valores de entrada	Valores de salida(esperados)	Valores obtenidos
La cantidad de semanas comprendidas entre el rango de fecha del Campo Inicio al Campo Fin no es válida por ser menor a 3 semanas. Los restantes campos válidos	False	False

Agregar Período al Curso 20

La cantidad de semanas(2) no pueden ser menor que 3

Nombre

Semestre I

Inicio

13/01/2020

Fin

25/01/2020

Inicio de turno de clases

08:00 AM

Entretiempo

10 minutos

Duración del turno de clases

90 minutos

Duración del receso

0 minutos

Aceptar

Resultado obtenido: Satisfactorio

Tabla 18 Caso de prueba H de particiones de equivalencia

Caso de prueba: I		
Valores de entrada	Valores de salida(esperados)	Valores obtenidos
La cantidad de semanas comprendidas entre el rango de fecha del Campo Inicio al Campo Fin no es válida por ser mayor a 24 semanas. Los restantes campos válidos	False	False

X
Agregar Período al Curso 20

La cantidad de semanas(25) no pueden ser mayor que 24

Nombre

Semestre I
▼

Inicio

06/01/2020
📅

Fin

27/06/2020
📅

Inicio de turno de clases

08:00 AM
🕒

Entretiempo

10 minutos
▼

Duración del turno de clases

90 minutos
▼

Duración del receso

0 minutos
▼

Aceptar

Resultado obtenido: Satisfactorio

Tabla 19 Caso de prueba I de particiones de equivalencia

Caso de prueba: J		
Valores de entrada	Valores de salida(esperados)	Valores obtenidos
Llenando los campos Inicio del turno de clases, Entretiempo y Duración del turno de clases y calculando las horas totales de duración de clases de un día, de modo que sobrepase el día. Los restantes campos válidos	False	False

Agregar Período al Curso 20

Las horas totales de clases(24.75) no pueden sobrepasar el día. Intente ajustar el inicio del turno de clases, el entretiempo, la duración del receso y/o la duración de los turnos

Nombre

Semestre I

Inicio

13/01/2020

Fin

27/06/2020

Inicio de turno de clases

11:30 AM

Entretiempo

15 minutos

Duración del turno de clases

120 minutos

Duración del receso

0 minutos

Aceptar

Resultado obtenido: Satisfactorio

Tabla 20 Caso de prueba J de particiones de equivalencia

3.5 Pruebas de aceptación

Las pruebas de aceptación son especificadas por el cliente, se centran en las características y funcionalidades generales del sistema que son visibles. Estas pruebas derivan de las HU que se han implementado como parte de la liberación del software. Una prueba de aceptación es como una caja negra. Cada una de ellas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección y toma de decisiones acerca de estas pruebas[29].

Con el objetivo de comprobar la aceptación del cliente con la solución desarrollada se realizaron un total de 3 iteraciones, obteniéndose los siguientes resultados:

Primera Iteración: Se detectaron 1 no conformidad (NC) por error de concepto y 3 NC de funcionalidad para un total de 4 NC.

Segunda Iteración: Se detectaron 1 NC de interfaz para un total de 1 NC.

Tercera Iteración: Se obtuvieron resultados satisfactorios al no detectarse no conformidades.

3.6 Conclusiones

Se aplicó la métrica CE proporcionando una medida cuantitativa de la calidad de la especificación de los requisitos, dando a conocer las ambigüedades existentes para su respectiva corrección. Se diseñaron y ejecutaron los casos de pruebas funcionales del sistema a través de las pruebas de caja blanca utilizando la técnica del camino básico para la obtención de un conjunto básico de caminos de ejecución de las sentencias del programa con los respectivos casos de pruebas por cada camino, además se realizó las pruebas de caja negra utilizando la técnica de particiones de equivalencia para examinar los valores válidos e inválidos de las entradas existentes en el software a través de casos de prueba. Se realizan las pruebas de aceptación con el cliente en las diferentes iteraciones, dando a conocer las no conformidades por errores encontrados corrigiéndose hasta la última iteración. Como resultado se corrigen los errores encontrados y se garantiza el control de la calidad en el desarrollo del software.

Conclusiones

Como resultado de la presente investigación se puede concluir que con la utilización del sistema desarrollado se logra mejorar el proceso de gestión de horarios docentes de la Universidad de Cienfuegos. Contribuyendo a mejorar el proceso docente educativo de la universidad.

Por lo tanto, se puede afirmar lo siguiente:

- ✓ Se describió el dominio del problema que es necesario conocer para entender el negocio del cliente.
- ✓ Se describió el flujo actual de los procesos.
- ✓ Se investigaron sistemas o softwares existentes en el ámbito internacional, nacional y local.
- ✓ Se seleccionaron las metodologías y/o tecnologías para desarrollar el sistema.
- ✓ Se diseñó un sistema informático que responde a las necesidades planteadas soportando el flujo actual de los procesos.
- ✓ Se validó el sistema realizando pruebas funcionales y pruebas de aceptación con el cliente corrigiendo satisfactoriamente los errores encontrados.

Recomendaciones

Para dar continuidad a esta investigación se recomienda:

- Implementar los dos primeros subprocesos mencionados en el segundo capítulo de esta investigación.
- Valorar la posibilidad de incorporar mecanismos de inteligencia artificial para la generación automática o semiautomática del balance de cargas.
- Agregar un mecanismo de notificación en caso de cambios que puedan surgir en los horarios.
- Codificar un conjunto mayor de restricciones para disminuir aun más el margen de error por parte de los planificadores.
- Incorporar la funcionalidad de acceder a los horarios de los locales y de los profesores.
- Incorporar la funcionalidad de acceder a los locales que están ocupados en el momento.

Referencias bibliográficas

- [1] J. F. Franco Baquero, E. M. Toro Ocampo, y R. A. Gallego Rendón, «http://www.scielo.org.co/scielo.php?script=sci_abstract&pid=S0122-34612008000200011&lng=en&nrm=iso&tlng=es», *Ing. Desarro.*, n.º 24, pp. 149-175, dic. 2008.
- [2] H. C. Fuentes González, R. de la Peña Silva, y M. R. Milán Licea, «La evaluación del proceso docente educativo como proceso participativo y no directivo», *Didascalia Didáctica Educ.*, vol. 1, n.º 2, pp. 39-52, 2010.
- [3] *Higiene de la actividad docente | ISBN 978-959-13-0408-7 - Libro*. Accedido: 25 de septiembre de 2023. [En línea]. Disponible en: <https://isbn.cloud/9789591304087/higiene-de-la-actividad-docente/>
- [4] R.- ASALE y RAE, «planificación | Diccionario de la lengua española», «Diccionario de la lengua española» - Edición del Tricentenario. Accedido: 25 de septiembre de 2023. [En línea]. Disponible en: <https://dle.rae.es/planificación>
- [5] L. A. Hernández-Leyva, S. Cleger-Tamayo, y R. I. Urquiza-Salgado, «Sistema informático para la elaboración y publicación del horario docente de la Universidad de Holguín», *Rev. Investig. Latinoam. En Compet. Organ.*, n.º febrero, feb. 2020, Accedido: 25 de septiembre de 2023. [En línea]. Disponible en: <https://www.eumed.net/rev/rilco/05/universidad-holguin.html>
- [6] Y. Reyes Fabregat, P. R. Quintero Rojas, y O. Fernández Gil, «Aplicacion Web para la confeccion de Horarios Docentes», bachelorThesis, 2009. Accedido: 27 de septiembre de 2023. [En línea]. Disponible en: https://repositorio.uci.cu/jspui/handle/ident/TD_2619_09
- [7] S. Cleger Tamayo, «Sistema Informático de apoyo a la Planificación Docente», Thesis, 2007. Accedido: 25 de septiembre de 2023. [En línea]. Disponible en: <http://repositorio.uho.edu.cu/xmlui/handle/uho/446>
- [8] «Estudiar el dominio del problema | Marco de Desarrollo de la Junta de Andalucía». Accedido: 6 de octubre de 2023. [En línea]. Disponible en: <https://www.juntadeandalucia.es/servicios/madeja/contenido/libro-pautas/175>
- [9] «SchoolTool Team OSC». Accedido: 6 de octubre de 2023. [En línea]. Disponible en: <https://sites.google.com/neric.org/osc>
- [10] «UniTime 4.7 Help». Accedido: 6 de octubre de 2023. [En línea]. Disponible en: <https://help.unitime.org/>
- [11] «Modelación Matemática y Solución del problema de Balance de Carga Docente en la Universidad de Cienfuegos. — Biblioteca Virtual de la UCF». Accedido: 5 de octubre de 2023. [En línea]. Disponible en: http://biblioteca.ucf.edu.cu/biblioteca/tesis/tesis-de-maestria/maestria-en-matematica-aplicada/ano-2014/Tesis_M%20Boris%20Perez%20Canedo.pdf/view?searchterm=Boris
- [12] «Sistema para la gestión y visualización de horarios docente — Biblioteca Virtual de la UCF». Accedido: 5 de octubre de 2023. [En línea]. Disponible en: <http://biblioteca.ucf.edu.cu/biblioteca/tesis/tesis-de-grado/ingenieria/informatica/curso-2012-2013/Renier%20Miranda%20Hernandez.pdf/view>
- [13] «Agile Unified Process - EcuRed». Accedido: 5 de octubre de 2023. [En línea]. Disponible en: https://www.ecured.cu/Agile_Unified_Process
- [14] E. H. Orallo, «El Lenguaje Unificado de Modelado (UML)».
- [15] «Modelo - Vista - Template», prezi.com. Accedido: 6 de octubre de 2023. [En línea]. Disponible en: <https://prezi.com/nd9ydrb01kqv/modelo-vista-template/>
- [16] «Visual Paradigm - EcuRed». Accedido: 12 de octubre de 2023. [En línea]. Disponible en: https://www.ecured.cu/Visual_Paradigm
- [17] «Axure - EcuRed». Accedido: 12 de octubre de 2023. [En línea]. Disponible en: <https://www.ecured.cu/Axure>
- [18] «Documentation for Visual Studio Code». Accedido: 3 de diciembre de 2023. [En línea]. Disponible en: <https://code.visualstudio.com/docs>
- [19] «Backend y Frontend, ¿Qué es y cómo funcionan en la programación?» Accedido: 7 de octubre de 2023. [En línea]. Disponible en: <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web>
- [20] «JavaScript | MDN». Accedido: 3 de diciembre de 2023. [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/Web/JavaScript>

- [21] «HTML5 - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web | MDN». Accedido: 3 de diciembre de 2023. [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/Glossary/HTML5>
- [22] «CSS | MDN». Accedido: 3 de diciembre de 2023. [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/Web/CSS>
- [23] «Bootstrap: qué es y cómo funciona este framework 【Usos】 ». Accedido: 3 de diciembre de 2023. [En línea]. Disponible en: <https://axarnet.es/blog/bootstrap#>
- [24] «Django», Django Project. Accedido: 3 de diciembre de 2023. [En línea]. Disponible en: <https://www.djangoproject.com/>
- [25] P. Londoño, «Qué es Python, para qué sirve y cómo se usa (+ recursos para aprender)». Accedido: 3 de diciembre de 2023. [En línea]. Disponible en: <https://blog.hubspot.es/website/que-es-python>
- [26] P. G. D. Group, «PostgreSQL», PostgreSQL. Accedido: 3 de diciembre de 2023. [En línea]. Disponible en: <https://www.postgresql.org/>
- [27] «Software - Requisitos». Accedido: 4 de diciembre de 2023. [En línea]. Disponible en: https://www.tutorialspoint.com/es/software_engineering/software_requirements.htm
- [28] «INGENIERIA DEL SOFTWARE UN ENFOQUE PRACTICO de PRESSMAN 978-607-15-0314-5». Accedido: 14 de noviembre de 2023. [En línea]. Disponible en: https://www.todostuslibros.com/libros/ingenieria-del-software-un-enfoque-practico_978-607-15-0314-5
- [29] IAN SOMERVILLE, *Ingeniería de Software*, 9na edición. PEARSON EDUCACIÓN. Accedido: 13 de octubre de 2023. [En línea]. Disponible en: http://artemisa.unicauca.edu.co/~cardila/IS__Libro_Sommerville_9.pdf
- [30] «Django_mtv.webp (1200×630)». Accedido: 12 de octubre de 2023. [En línea]. Disponible en: https://espifreelancer.com/images/Django_mtv.webp
- [31] @Espifreelancer, «Que es el patrón MTV (Model Template View)», EspiFreelancer. Accedido: 12 de octubre de 2023. [En línea]. Disponible en: <https://espifreelancer.com/mtv-django.html>
- [32] P. G. MANUEL JOSE, *Bases de datos relacionales y modelado de datos*. Ediciones Paraninfo, S.A.
- [33] R. Camps Paré, L. A. Casillas Santillán, D. Costal Costa, M. Gibert Ginestà, C. Martín Escofet, y O. Pérez Mora, «Bases de datos: Software libre», 2007, Accedido: 13 de octubre de 2023. [En línea]. Disponible en: <http://libros.metabiblioteca.org/handle/001/264>
- [34] «Particion de Equivalencia». Accedido: 16 de noviembre de 2023. [En línea]. Disponible en: <https://www.fing.edu.uy/inco/cursos/ingsoft/pis/memoria/dvd01/experiencia2005/MUM/protest/guias/guias/particionE.htm>

Anexos

Anexo 1 Modelo oficial del P4



P-4

Curso Diurno
Semestre: Semestre I
Carrera: Informática

Actualizado el 24 de noviembre de 2023 a las 14:55
Grupo: 1.1
Curso: Curso 23

Facultad: Facultad de Ingenierías
Año: Primer año

1: 20/02/2023 al 25/02/2023							2: 27/02/2023 al 04/03/2023							3: 06/03/2023 al 11/03/2023							4: 13/03/2023 al 19/03/2023							5: 20/03/2023 al 25/03/2023							6: 27/03/2023 al 01/04/2023							
L	M	M	J	V	S		L	M	M	J	V	S		L	M	M	J	V	S		L	M	M	J	V	S		L	M	M	J	V	S		L	M	M	J	V	S		
7: 03/04/2023 al 08/04/2023							8: 10/04/2023 al 15/04/2023							9: 17/04/2023 al 22/04/2023							10: 24/04/2023 al 29/04/2023							11: 01/05/2023 al 06/05/2023							12: 08/05/2023 al 13/05/2023							
13: 15/05/2023 al 20/05/2023							14: 22/05/2023 al 27/05/2023							15: 29/05/2023 al 03/06/2023							16: 05/06/2023 al 10/06/2023							17: 12/06/2023 al 17/06/2023							18: 19/06/2023 al 24/06/2023							

SMB	ASIGNATURA	PROFESOR

ACTIVIDAD	SMB

Aula: C-2

J Colectivo de año

Decano

Anexo 2 Pruebas unitarias

```
from django.test import TransactionTestCase

# Create your tests here.

import datetime
from .forms import PeriodForm
from .models import Course, PeriodName

class PeriodFormTestCase(TransactionTestCase):

    def __init__(self, methodName: str = "runTest") -> None:
        super().__init__(methodName)

    def setUp(self) -> None:
        self.period_name, _ = PeriodName.objects.get_or_create(name='Semestre I')
        self.course, _ = Course.objects.get_or_create(name='Curso 20',
                                                    start=datetime.date(
                                                        year=2020,
                                                        month=1,
                                                        day=6),
                                                    end=datetime.date(
                                                        year=2020,
                                                        month=12,
                                                        day=19)
                                                    )

        self.data = {
            'name': self.period_name.name,
            'course': self.course.id,
            'start': datetime.date(year=2020, month=1, day=13),
            'end': datetime.date(year=2020, month=6, day=27),
```

```

        'start_of_classtime': datetime.time(hour=8, minute=0),
        'halftime': 10,
        'classtime_duration': 90,
        'duration_of_recess': 0
    }
    return super().setUp()

def test_valid_name(self):
    data = self.data
    form = PeriodForm(self.course.id, data=data)
    self.assertTrue(form.is_valid())

def test_invalid_name(self):
    data = self.data
    data['name'] = 'Semestre i3jnrj3rf 3i4jnj34nijn3f i3ijj3jrnfij3rnf'
    form = PeriodForm(self.course.id, data=data)
    self.assertFalse(form.is_valid())

def test_invalid_start_outside_course_range(self):
    data = self.data
    data['start'] = datetime.date(year=2019, month=1, day=13)
    form = PeriodForm(self.course.id, data=data)
    self.assertFalse(form.is_valid())

def test_invalid_start_sunday(self):
    data = self.data
    data['start'] = datetime.date(year=2020, month=1, day=19)
    form = PeriodForm(self.course.id, data=data)
    self.assertFalse(form.is_valid())

def test_invalid_end_outside_course_range(self):
    data = self.data
    data['start'] = datetime.date(year=2020, month=9, day=7)
    data['end'] = datetime.date(year=2021, month=1, day=9)
    form = PeriodForm(self.course.id, data=data)
    self.assertFalse(form.is_valid())

def test_invalid_end_sunday(self):
    data = self.data
    data['end'] = datetime.date(year=2020, month=6, day=21)
    form = PeriodForm(self.course.id, data=data)
    self.assertFalse(form.is_valid())

def test_invalid_start_greater_than_end(self):
    data = self.data
    data['start'] = datetime.date(year=2020, month=4, day=13)
    data['end'] = datetime.date(year=2020, month=1, day=13)
    form = PeriodForm(self.course.id, data=data)
    self.assertFalse(form.is_valid())

def test_invalid_number_of_weeks_less_than_3(self):

```

```
data = self.data
data['end'] = datetime.date(year=2020, month=1, day=25)
form = PeriodForm(self.course.id, data=data)
self.assertFalse(form.is_valid())

def test_invalid_number_of_weeks_more_than_24(self):
    data = self.data
    data['start'] = datetime.date(year=2020, month=1, day=6)
    form = PeriodForm(self.course.id, data=data)
    self.assertFalse(form.is_valid())

def test_invalid_hours_of_a_school_day_more_than_24(self):
    data = self.data
    data['start_of_classtime'] = datetime.time(hour=11, minute=30)
    data['halftime'] = 15
    data['classtime_duration'] = 120
    form = PeriodForm(self.course.id, data=data)
    self.assertFalse(form.is_valid())
```